NASA Contractor Report 195291

1N-08 307 P-76

# Computer Code for Controller Partitioning With IFPC Application

A User's Manual

Phillip H. Schmidt and Asim Yarkhan University of Akron Akron, Ohio

March 1994

Prepared for Lewis Research Center Under Grant NAG3-11467



(NASA-CR-195291) COMPUTER CODE FOR CONTROLLER PARTITIONING WITH IFPC APPLICATION: A USER'S MANUAL Final Report (Akron Univ.) 76 p

N94-27414

unclas

G3/08 0000307

		क् <sub>र</sub> वर्ष	
			÷
			•
:	3		•
			•

#### **ABSTRACT**

This is a user's manual for the computer code for partitioning a centralized controller into decentralized subcontrollers with applicability to Integrated Flight/Propulsion Control (IFPC). Partitioning of a centralized controller into two subcontrollers is described and the algorithm on which the code is based is discussed. The algorithm uses parameter optimization of a cost function which is described here. The major data structures and functions are described. Specific instructions are given. The user is led through an example of an IFPC application.

## User's Manual — Table of Contents

1.	Introduction pa	ıge	1
2.	THE COST FUNCTION pa	ige	4
3.	THE PARTITIONING ALGORITHM pa	ige	7
4.	Major Data Structures and Variables pag	ge :	12
5.	Instructions to User pag	ge :	16
6.	Example of Controller Partitioning page	ge	19
	References pag	ge :	28
	APPENDICES		
	I. DEVELOPMENT OF COST AND GRADIENT pag	ge	29
	II. SHORT DESCRIPTIONS OF MATRIXX USER-DEFINED FUNCTIONS page	ge	41
	III. CONTROLLER PARTITIONING CODE pag	ge	48
	IV. Data for Example pag	ge	65

### Acknowlegment

The authors wish to express their appreciation to Dr. Sanjay Garg of NASA Lewis Research Center for his excellent technical guidance and support during the project described here.

#### 1. Introduction

Large interconnected systems such as the flight/propulsion systems of modern aircraft often exhibit significant coupling between the various subsystems. One example of such a system is the Short Take-Off and Landing (STOL) aircraft wherein the forces and moments generated by the propulsion system provide control and maneuvering capabilities for the aircraft at low speeds. This strong coupling suggests that a centralized control design be used, however, a centralized controller which is designed for the integrated plant considering all the interconnections between the flight and propulsion subsystems may be of high order and may be difficult to implement and validate. Specifically, in aircraft design it is the responsibility of the engine manufacturer to ensure that the propulsion system will provide the desired performance when installed in the aircraft. The engine manufacturer thus needs a separate engine controller to be able to perform extensive testing to assure adequate performance and integrity in the presence of operational and safety limits. This requirement suggests the need for decentralized implementation of Integrated Flight/Propulsion Control (IFPC) systems.

One approach to integrated control design which combines aspects of centralized and decentralized control design approaches is currently being developed at the NASA Lewis Research Center [1]. This approach consists of first designing a centralized controller, so that all subsystem interconnections are accounted for in the initial design stage, and then partitioning the centralized controller into separately implementable decentralized subcontrollers for individual subsystems. Here, partitioning means representing the high-order centralized controller with two or more lower order subcontrollers which have input/output intercoupling such that the overall control law obtained on assembling the subcontrollers closely approximates the input/output behavior of the centralized controller.

The computer code described in this user's manual is designed specifically for IFPC application and the notation and terminology used here reflects that application. The software described here uses a parameter optimization method to match the performance of a centralized controller with a partitioned controller consisting of two decentralized subcontrollers for the flight and propulsion systems. This matching will be subject to certain subsystem design requirements. The structure is shown in Fig. 1.1, with optional feedback paths indicated by dotted lines.

In the decentralized, hierarchical controller partitioning structure shown in Fig. 1.1,

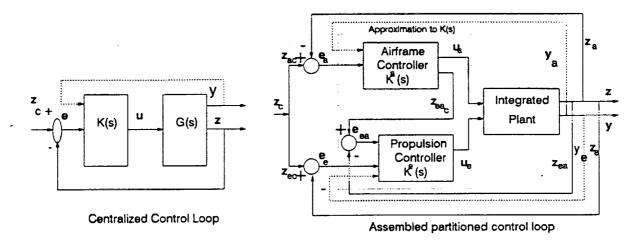


Figure 1.1 Controller Partitioning

the subscripts and superscripts "a" and "e" refer to airframe and propulsion (engine) quantities, respectively, and the subscript "c" refers to commanded quantities. The interface variables  $\mathbf{z}_{ea}$  represent propulsion system quantities that affect the airframe, such as propulsive forces and moments. The structure is hierarchical in that the airframe (flight) controller produces commands for the engine controller via the interface variable  $(\mathbf{z}_{ea})$  which are tracked by the propulsion subsystem.

Such a control structure allows the engine manufacturer to evaluate the engine subsystem performance independently of the airframe control and to verify that the engine subsystem will provide the desired performance when installed in the airframe. In general there are practical constraints on the achievable bandwidth of  $\mathbf{z}_{ea}$  tracking for the engine subcontroller. A lower bound on the  $\mathbf{z}_{ea}$  command tracking bandwidth is based on achieving the desired performance for the integrated system, while an upper bound is imposed by actuator limits and robustness requirements to high frequency modeling uncertainties.

The software discussed here refers to the structure described above. The variables are named according to the convention given above. The parameters in this optimization process are entries in the state-space representations of the subcontrollers. These parameters are bounded so as to maintain subcontroller (open-loop) stability. An assumption made in the formulation is that the plant has no direct feedthrough from control inputs, i.e. the plant "D" matrix is zero. This simplifies the determination of the cost function and its gradient.

One feature of the software is that the user may separately optimize the airframe

controller for a fixed engine controller or optimize the engine controller for a fixed airframe controller. The main alternative is to jointly optimize both although separate optimization is demonstrated in the example.

This user's manual is organized as follows. Section 2 briefly describes the cost function which is the objective to be minimized. The partitioning algorithm is described in section 3. There is also a brief description here of the interrelation among User-defined Functions (UDFs) so the user who wishes to change the cost function or partitioning structure will know which UDFs must be changed. Section 4 describes the data structures needed for carrying out partitioning using the MATRIX x programming language. This section also contains a description of the major data structures and variables used. Section 5 has a brief outline of the procedure for using the software. Section 6 contains a detailed example which exercises the algorithm showing applications of its options.

Appendix I contain a detailed discussion of the parameterization, the cost function and the gradient evaluation as they are implemented in the software. Appendix II contains short descriptions of the UDFs which implement the partitioning algorithm. Appendix III contains fully-documented source code for partitioning. Appendix IV contains the data file, INIT.DAT, and partitioned subcontrollers, [SKA\_OPT, SKE\_OPT] for the example in section 6.

#### 2. THE COST FUNCTION TO BE MINIMIZED

With reference to Figure 1.1, the partitioning problem can be stated as follows:

Given a centralized controller with transfer matrix K(s) and a specification of the partitioning structure of controller inputs and outputs, i.e.

$$\begin{bmatrix} \mathbf{u}_a \\ \mathbf{u}_e \end{bmatrix} = K(s) \begin{bmatrix} \mathbf{e}_a \\ \mathbf{e}_e \\ \mathbf{y}_a \\ \mathbf{y}_e \end{bmatrix}$$
(2.1)

where  $\mathbf{u}_a \in R^{k_a}$ ,  $\mathbf{u}_e \in R^{k_e}$ ,  $\mathbf{e}_a \in R^{m_a}$ ,  $\mathbf{y}_a \in R^{l_a}$ ,  $\mathbf{e}_e \in R^{m_e}$  and  $\mathbf{y}_e \in R^{l_e}$ ; a choice of interface variables  $\mathbf{z}_{ea} \in R^{p_{ea}}$ ; a plant with transfer matrix  $\widehat{G(s)}$  of the form

$$\widehat{G(s)} = \begin{bmatrix} G(s) \\ G_{ea}(s) \end{bmatrix} \quad \text{with} \quad \begin{bmatrix} \mathbf{z}_a \\ \mathbf{y}_a \\ \mathbf{z}_e \\ \mathbf{y}_e \end{bmatrix} = G(s) \begin{bmatrix} \mathbf{u}_a \\ \mathbf{u}_e \end{bmatrix} \text{ and } \mathbf{z}_{ea} = G_{ea}(s) \begin{bmatrix} \mathbf{u}_a \\ \mathbf{u}_e \end{bmatrix}$$

find subcontrollers with stable transfer matrices  $K^a(s)$  and  $K^e(s)$ , i.e.

$$\begin{bmatrix} \mathbf{u}_a \\ \mathbf{z}_{ea_e} \end{bmatrix} = K^a(s) \begin{bmatrix} \mathbf{e}_a \\ \mathbf{y}_a \end{bmatrix} \quad \text{and} \quad \mathbf{u}_e(s) = K^e(s) \begin{bmatrix} \mathbf{e}_{ea} \\ \mathbf{e}_e \\ \mathbf{y}_e \end{bmatrix};$$

where  $\mathbf{e}_{ea} = \mathbf{z}_{ea_c} - \mathbf{z}_{ea}$ , so that the closed-loop performance with the sub-controllers closely matches that with the centralized controller within the requirements of the subsystem.

The particular subsystem constraint for IFPC application is that the engine subcontroller  $K^{\epsilon}(s)$  should have the structure of a command tracking controller for the interface variable commands  $\mathbf{z}_{\epsilon a_{\epsilon}}$ .

The cost function is formulated to reflect the difference between the centralized and partitioned controllers. The state space representations of the subcontrollers  $K^a(s)$  and  $K^e(s)$  are parameterized and the cost function is minimized over those parameters denoted as a vector  $\mathbf{p}$ . The formulation of this parameterization is discussed in Appendix I. Stability robustness may be achieved through the use of optional (user-provided) weighting matrices and a normalization function in determining the partitioning cost. Specific details

concerning the cost function, the parameters involved and the evaluation of the gradient of the cost are contained in Appendix I.

The cost function  $f(\mathbf{p})$  is the sum of the performance cost,  $f_0(\mathbf{p})$ , and an additional cost of tracking the airframe-to-engine commands,  $f_1(\mathbf{p})$ ,  $f(\mathbf{p}) = f_0(\mathbf{p}) + f_1(\mathbf{p})$ .

The performance cost,  $f_0$ , is the  $H_2$  norm of the weighted (and possibly normalized) difference of transfer matrices for the centralized and partitioned controllers

$$f_{0}(\mathbf{p}) = \int_{0}^{\infty} \frac{1}{N_{\text{PERF}}(\omega)} \operatorname{tr} \left[ \left( W_{o}(j\omega)(K(j\omega) - \widetilde{K(\mathbf{p})}(j\omega))W_{i}(j\omega) \right)^{*} \right] \left( W_{o}(j\omega)(K(j\omega) - \widetilde{K(\mathbf{p})}(j\omega))W_{i}(j\omega) \right) d\omega \quad (2.2)$$

where K is the transfer matrix from the  $\begin{pmatrix} \mathbf{e} \\ \mathbf{y} \end{pmatrix}$  inputs to the  $\mathbf{u}$  outputs for the centralized controller, and  $\widetilde{K}(s)$  is the transfer matrix of an "equivalent" centralized controller (having the same input/output structure as K) obtained by assembling the partitioned subcontrollers using appropriate plant information. Details of the state space representation for  $\widetilde{K}(s)$  are given in Appendix I.

We are using the  $H_2$  norm of the weighted difference between the transfer matrices for the centralized controller and the equivalent heirarchically partitioned subcontrollers as will be described in Appendix I. Since this difference must be strictly proper in order to apply this norm, it is reasonable for the D matrices for the centralized and partitioned controllers to be the same. Thus it may be desirable to fix the values of  $D_{aa}^a$ ,  $D_{aya}^a$ ,  $D_{ee}^e$ , and  $D_{eye}^e$  (as described in Appendix I) to values determined directly by the centralized controller. This is one of the options available in "fixing the D-parameters".

 $W_i(j\omega)$  and  $W_o(j\omega)$  are optional input and output weighting matrices,  $N_{\rm PERF}(\omega)$  is an optional scalar normalization function. For example, the weighting

$$W_i(s) = G(s) \left(I + K(s)G(s)\right)^{-1}$$

has been shown by Dale Enns [2] to lead to stability robustness for the partitioned system provided that the centralized system has this property. Other weighting and normalization will be discussed with the example.

 $f_1(\mathbf{p})$  is the cost of tracking the  $\mathbf{z}_{ea_c}$  command generated by the airframe subcontroller for the engine subcontroller. This cost minimizes the difference between the transfer matrices for the responses to the  $\mathbf{z}_{a_c}$  command of  $\mathbf{z}_{ea_c}$  using the partitioned controller and

zea using the centralized controller.

$$f_1(\mathbf{p}) = \int_0^\infty \sum_i \frac{\lambda_i}{N_{\text{TRACK}_i}(\omega)} \left( ||T_{\text{cent}}^i(j\omega) - \widehat{T}^i(\mathbf{p})(j\omega)||_2 \right)^2 d\omega$$
 (2.3)

 $T_{\text{cent}}^i$  is the transfer function vector from the airframe commands  $\mathbf{z}_{a_c}$  to the  $i^{\text{th}}$  interface variable  $\mathbf{z}_{ea}^i$  with the centralized controller.  $\widehat{T}^i$  is the  $i^{\text{th}}$  row of the transfer function matrix  $\widehat{T}$  from the airframe commands  $\mathbf{z}_{a_c}$  to the interface variables as commanded by the partitioned airframe controller,  $\mathbf{z}_{ea_c}$ , with the partitioned subcontrollers.  $\lambda_i$  is a scalar weighting which determines the influence of  $f_1$  on the total cost and  $N_{\text{TRACK}_i}(\omega)$  are (optional) scalar normalization functions.  $||\cdot||_2$  denotes the Euclidean norm of the row vector. Here, one may use the normalizations  $N_{\text{TRACK}_i} = ||T_{\text{cent}}^i||_2^2$  to provide adequate scaling for this cost. The parameters  $\lambda_i$  provide weighting for the contribution of the tracking cost to the total cost. It was shown in [3] that manipulating  $\lambda_i$  provides an indirect means for maintaining reasonable bounds on the  $\mathbf{z}_{ea_c}$  command tracking bandwidth.

It may be required that the engine subsystem be proper, a condition which would be violated if  $D_{eea}^e$  (described in Appendix I) is nonzero. As a result of the optimization process,  $D_{eea}^e$  may become large. This possibility is removed by "fixing  $D_{eea}^e = 0$ " when the option is presented while running the code.

# 3. THE PARTITIONING ALGORITHM AND ITS IMPLEMENTATION

The objective is to minimize the cost  $f(\mathbf{p}) = f_0(\mathbf{p}) + f_1(\mathbf{p})$  as described above where the parameters  $\mathbf{p}$  are certain entries in the state space representation matrices for  $K^a(s)$  and  $K^c(s)$  (denoted SKA and SKE in the code).

The fixed data used by the algorithm are state-space representations for the plant transfer matrix G(s), the centralized controller K(s), the (optional) weighting matrices  $W_i(s)$  and  $W_o(s)$  (denoted as SP, SC, SWI and SWO respectively in the code), as well as a partitioning structure for the numbers of controller inputs (airframe, MA; and engine, ME), numbers of outputs (airframe, KA; and engine, KE), numbers of plant measurements (airframe, LA; and engine, LE) and numbers of airframe to engine subcontrollers interface variables (PEA). The control designer may also introduce normalization functions (N\_PERF and N\_TRACK) for the performance and tracking costs. Examples of normalizations are given with the example in Section 6. The user must also enter values of the tracking weight parameters  $\lambda_i$  which determine the relative contribution of  $f_1$  to the total cost.

The algorithm incorporates the Broyden-Fletcher-Shanno-Goldfarb (BFGS) quasi-Newton method to select directions of search for Fletcher's inaccurate linesearch, see [4]. This iterative method requires the calculation of the combined cost and its gradient for the parameters p as referred to above. It uses successive gradients to build up an approximation to the inverse Hessian matrix. Moreover, the inaccurate linesearch assures an adequate reduction in the cost function at each step without using excessive effort searching for a minimum far away from the ultimate solution. In this way, subsequent search steps are successively closer to those generated by Newton's Method and convergence is accelerated as the iterations proceed.

The flow of the parameter optimization algorithm for controller partitioning is shown in Figure 3.1. The main steps in the algorithm are:

1. The initial partitioning is obtained by applying the stepwise procedure described in [5]. Special attention is paid to obtaining reasonably low-order subcontrollers which are stable and satisfy the  $\mathbf{z}_{ea}$  command-tracking requirement. The initial state-space representations for the transfer matrices  $K^a(s)$  and  $K^e(s)$  are denoted as S\_KA and

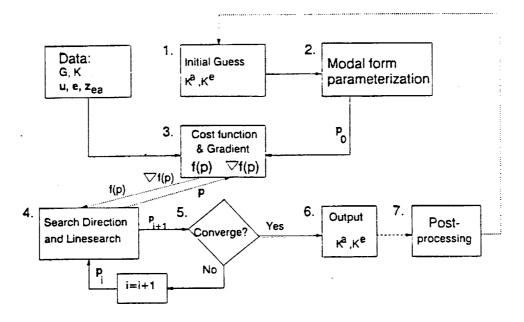


Figure 3.1 Flowchart for Partitioning Optimization Algorithm

S\_KE; note that these are different from the working representations SKA and SKE described next.

- 2. The initial partitioning is converted to a "minimal parameter" form (with state-space representations SKA and SKE respectively) and used to generate an initial value of the parameter vector P.I. This form is described in Appendix I.
- 3. The initial (as well as any subsequent) value of the parameter vector is passed to a function which determines the state-space representation for the equivalent partitioned controller and calculates the combined cost, f (denoted FP in the code). The gradient (denoted as DFDP) is also computed analytically by the procedure described in Appendix I.
- 4. The BFGS method uses the current gradient in conjunction with previous information to generate a direction of search. The Fletcher inaccurate linesearch is carried out using the cost and gradient calculated at each parameter vector to predict a new parameter vector until one is found which yields a sufficient reduction in both the cost function and the size of the gradient. The new point is denoted as P\_I1. This linesearch is constrained so as to maintain stability of the subcontrollers.

- 5. At the end of the linesearch, the new parameter vector and cost (P\_II and FP\_II) are compared to the values at the beginning of the linesearch (P\_I and FP\_I) as a check on convergence. If the maximum change in all the parameters is less than a user-specified value and the change in the total cost is less than another value (MAX(P\_II-P\_I) < EPSILON and ABS(FP\_II-FP\_I) < DELTA) then convergence is declared and the iteration ceases. If in addition, the maximum absolute value of the partial derivatives is less than a user-specified tolerance (ABS(DFDP) < ETA), this is also noted. If the number of iterations exceeds ITER or the function value is sufficiently reduced (FP < FMIN), then the procedure stops with an appropriate message. If the convergence test fails, the algorithm proceeds to update the information used to determine the direction of search and to use the most recent cost and gradient values to generate a new direction of search and carry out the linesearch via steps 3. and 4.
- 6. The output of the algorithm is the state-space representation for subcontrollers (denoted SKA\_OPT and SKE\_OPT) which minimize the cost function  $f(\mathbf{p})$  within the convergence criteria.
- 7. These subcontroller transfer matrices have the same orders  $n_a$  and  $n_e$  as the initial partitioning. Controller reduction can be performed on these "optimal subcontrollers" and the process of optimization can be repeated on the "new initial partitioning".

The algorithm is implemented in MATRIX<sub>X</sub> using a set of functions which are referred to in MATRIX<sub>X</sub> parlance as User-Defined Functions (UDFs). A glossary of variables and UDFs follows in Section 4. More complete descriptions of the UDFs appear in Appendix II and an annotated version of the code appears in Appendix III. The flow of the MATRIX<sub>X</sub> partitioning code as illustrated in Figure 3.2 follows:

- a. The function START is called with input LAMBDA (required tracking cost weight scalar or vector) and STOP (optional stopping criteria vector).
  - i. A file (INIT.DAT) is read to acquire the fixed data and initial partitioning (S\_KA and S\_KE) referred to above along with a three dimensional vector (FRQ) which gives the left and right end points as well as the number of logarithmically placed points in the interval over which numerical integration takes place.
  - ii. START calls the routine MODL to put the initial partitioning into an appropriate

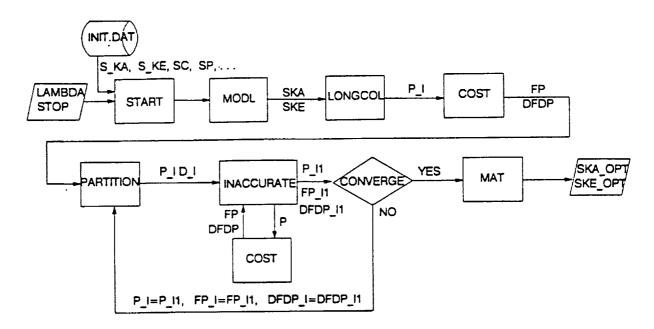


Figure 3.2 Flowchart for the MATRIXX Partitioning Code

form to serve as a parameterization for the subcontrollers, i.e.  $SKA = MODL(S\_KA)$  and  $SKE = MODL(S\_KE)$ .

- iii. The function LONGCOL transforms these state-space representations for the subcontrollers into a parameter vector, i.e. P.I = LONGCOL( SKA, SKE ). This function along with its "inverse" MAT ([ SKA, SKE ] = MAT( P.I )) are used throughout the code to transform between state-space representations for subcontrollers and a parameter vector. START also determines constants which will be used by the other functions and stores them in CONST.DAT .
- iv. START then calls COST to calculate the initial cost (FP) and gradient (DFDP).
- b. START calls PARTITIO with the convergence criteria (STOP) as input. PARTITIO is the main routine which
- c. generates the search direction (DI),
- d. calls the function INACCURATE which carries out Fletcher's inaccurate linesearch (bracketing/sectioning) using cost function, F(P\_I+ALPHA\*DI), and gradient values generated by the function COST. While INACCURATE is running, the user will see displayed first the bracketing interval (AL, ALPR) and function and derivative values (FAL, FPAL; FALPR, FPALPR) then the sectioning interval (A, B) and corresponding

function and derivative values (FA, FPA; FB, FPB). These give the user some sense of how rapidly the linesearch is progressing, but they can easily be removed from the code if desired.

- e. Convergence/stopping conditions are checked by the function CONVERGE and steps c. e. are repeated until they are satisfied. During these major iterations, the last twenty cost values are plotted.
- f. The output of START is the final partitioning (SKA\_OPT, SKE\_OPT). At this point, the program ends.

The user may do a posteriori analysis such as order reduction on the subcontroller statespace representations (SKA\_OPT, SKE\_OPT) and start the procedure again with new data in the file INIT.DAT.

If the user wishes to modify the partitioning structure, changes will be necessary in the START, COST, LONGCOL and MAT routines. Different constants must be calculated and stored by START. Different formulations for the state space representation of  $\widetilde{K}(s)$  and  $\widehat{T}(s)$  must be coded in COST and new formulations for the gradient must be generated using the procedure described in Appendix I. Furthermore, the conversions between [SKA,SKE] and p by LONGCOL and MAT must be rewritten.

If a different formulation of the cost function is used, then only the portion of the COST UDF where the cost and gradient are computed must be changed. The algorithm requires a gradient with each evaluation of the cost. The new gradient may be the most difficult change to make.

#### 4. REFERENCE TO MAJOR DATA STRUCTURES AND VARIABLES

Most of the input to the program is provided through the MATRIX x data file INIT.DAT. The following data structures are mandatory to the running of the program and must be provided in INIT.DAT prior to running the program.

INIT.DAT — Mandatory Data

SP, NP — the state space representation and the order for the integrated plant in system matrix form

$$SP = \begin{pmatrix} AP & BP \\ CP & DP \end{pmatrix}.$$

SC, NSC — the state space system matrix for the centralized controller, and its order.

S\_KA, NS\_KA — the state space system for an initial "guess" at the airframe controller, and its order.

S\_KE, NS\_KE — the state space system for an initial "guess" at the engine controller and its order.

PEA — the number of interface variables from flight controller to the engine controller.

FRQ — a vector of the form [FRQ(1);FRQ(2);FRQ(3)] where FRQ(3) logarithmically placed frequency points over the interval FRQ(1)  $< \omega <$  FRQ(2) are used in the numerical integration for determining the costs. The number of points must be odd because of the numerical integration rule used.

The following data structures are optional and may be entered in INIT.DAT if desired. Indexing rules of MATRIX x do not allow the index zero or empty vectors. Thus, if some of the optional quantities are absent or have value zero, the code will place dummy variables in appropriate matrices and set corresponding size variables to nonzero quantities (usually one).

INIT.DAT — Optional Data — if absent, the indicated default values are set by the code.

SWI, NWI — state space system for input weighting of the difference between centralized and assembled partitioned controllers, and its order. If absent, the code sets SWI = an identity matrix of size (MA + ME + LA + LE + 1) and NWI = 1.

- SWO, NWO state space system for output weighting of the difference between centralized and assembled partitioned controllers, and its order. If absent, the code sets SWO = an identity matrix of size (KA + KE + 1) and NWO = 1.
- LA number of integrated plant measurements to the airframe controller. If absent, the code sets LA = 1; appropriate zero entries are introduced in SP, SC, and SKA.

  This allows for the case where there are no measurements fed from the plant to the airframe controller.
- LE number of integrated plant measurements to the engine controller. If absent, the code sets LE = 1; appropriate zero entries are introduced in SP, SC, and SKE. This allows for the case where there are no measurements fed from the plant to the engine controller.
- NPERF normalization vector of size (FRQ(3) × 1) for the performance cost. If absent, the code sets NPERF to a vector of ones.
- NTRACK normalization matrix of size (FRQ(3) × PEA) for the tracking cost. If absent, the code sets NTRACK to a matrix of ones.
- STABIL a necessarily negative parameter which is used to guarantee that all eigenvalues of the subcontrollers have negative real parts for stability. If absent, the code sets  $STABIL = -10^{-9}$ .

The major constants used within the code are created by the execution of START and stored in the file CONST.DAT.

- MA, ME number of airframe and engine controller inputs.
- LA, LE number of integrated plant measurements to airframe and engine controllers.
- KA, KE number of airframe and engine controller outputs.
- PEA number of intermediate commands from airframe controller to engine controller.
- SP, NP state space system for the integrated plant, with its order.
- $S_KA$ ,  $NS_KA$  state space system for the initial airframe controller  $(K^a)$  in modified

- modal form (or in original form if the system is to be held fixed) along with its order.
- S\_KE, NS\_KE state space system for the initial engine controller  $(K^e)$  in modified modal form (or in original form if the system is to be held fixed) along with its order.
- SK, NK state space system for centralized controller (K), with its order.
- SG, NG state space system for transfer matrix from  $\mathbf{z}_{a_c}$  to  $\mathbf{z}_{ea}$  ( $T_{cent}$ ) using the centralized controller, with its order.
- WEIGHT a vector containing weights to use with Simpson's integration rule; depends on FRQ(3) for the number of points at which integration is to take place.
- OMEGA the vector of logarithmically placed frequency points at which sampling is to be done; determined by FRQ.
- AORE flag to indicate whether the airframe controller (1) or the engine controller (2) or neither (0) is held fixed during the optimization process.
- FIXD flag to indicate whether the 'D' matrices are held fixed; if FIXD = 0 none are fixed, if FIXD = 1 or 2 the DAA, DAYA, DEE and DEYE are held fixed during the optimization process and (if FIXD = 1) then DEEA and DEAE are variables or (if FIXD = 2) then DEEA is set to a zero matrix and DEAE is a variable matrix; if FIXD = 3 then only DEEA is set to a zero matrix and all the remaining D's are variable matrices. In any case DAA, DAYA, DEE and DEYE are saved in CONST.DAT.
- STABIL a necessarily negative parameter which is used to guarantee that all eigenvalues of the subcontrollers have negative real parts for stability. If absent, the code sets  $STABIL = -10^{-9}$ .

The outputs of interest to the program are kept in the file INTER.DAT, which contains a history of the optimization process and information which can be used to restart the program (after a crash or after intentionally stopping it) if desired.

#### INTER.DAT contains

- P.I the last point (parameter vector) that met the inaccurate linesearch minimization criteria, that is, the  $i^{th}$  point. P.I is a column vector consisting of the successive  $\alpha, \beta$  values in the  $2 \times 2$  blocks of  $A^a$  and  $A^e$ , the successive columns of  $B^a_{aa}$  after the first, the successive columns of  $C^a$ , the successive columns of  $D^a$  corresponding to the  $\mathbf{z}_{ea}$  outputs, the successive columns of  $B^e$  after the first, the successive columns of  $C^e_{ee}$  and the successive columns of  $D^e$  corresponding to the  $\mathbf{e}_{ea}$  inputs in this order. See Appendix I for a more complete discussion of the parameter vector as it is related to the cost function.
- JH0 the complete history of the cost of partitioning through the  $i^{th}$  iteration.
- JH1 the complete history of the cost of tracking through the  $i^{th}$  iteration.
- FH the complete history of the total cost, FP = FP0 + FP1 through the  $i^{th}$  iteration.
- GRADO the gradient of the partitioning part of the cost function,  $f_0$ , at the  $i^{th}$  iteration.
- GRAD1 the gradient of the tracking part of the cost function,  $f_0$ , at the  $i^{th}$  iteration.
- GRADI the total gradient of the cost function at the  $i^{th}$  iteration.
- LAMBDA the PEA×1 vector which weights the contribution of the airframe to engine command tracking cost in the total cost function.
- HI the inverse Hessian matrix being used during the BFGS optimization process.
- FX keeps track of which parameters corresponding to  $A^a$  and  $A^e$  are at the stability bound.

Refer to the cost section of Appendix I for the structures of SP, SK, SKA, SKE and SG.

#### 5. Instructions to the User

The following instructions are intended to be a quick introduction to the code. They do not attempt to explain the details of what is occurring during the execution. More thorough documentation of the MATRIXX functions is available in the documented code contained in Appendix III.

- Use MATRIX<sub>X</sub> to construct the data file INIT.DAT described in the preceding section. The initial approximation to the subcontroller state-space matrices S\_KA and S\_KE can be produced by the procedure described in [5] or may come from a previous application of the partitioning software. If any of the optional data are not present in INIT.DAT then the code will produce the defaults indicated earlier.
- Start MATRIXX and type

#### DEFINE 'START.MTX'

to activate the code. To execute START you must enter a value of LAMBDA, a weighting of the tracking cost relative to the total cost. LAMBDA can be a PEA  $\times$  1 vector whose entries individually weight the  $\mathbf{z}_{ea}$  output responses to the total  $\mathbf{z}_{a_e}$  inputs. If all the weights are to be the same then a scalar may be entered. This variable emphasizes the degree to which the tracking cost will affect the total cost.

Optionally, the vector STOP of stopping conditions can be defined.

STOP = [EPSL; DELTA; ETA; ITER; FMIN].

The program stops if the following criteria are met:

the maximum change in the parameters MAX(|P\_J-P\_J1|) < EPSL and

the change in the cost |FPJ-FPJ1| < DELTA and

the norm of the gradient |DFDP\_I| < ETA or

the number of major iterations I > ITER or

the cost FP\_I1 < FMIN

If STOP is not entered, the following values are set by the code STOP = [EPSL; DELTA; ETA; ITER; FMIN] =  $[10^{-9}; 10^{-9}; 10^{-9}; 100; 0.1]$ .

• Now the program can be executed by entering

[SKA\_OPT,SKE\_OPT]=START(LAMBDA, STOP) or [SKA\_OPT,SKE\_OPT]=START(LAMBDA).

The choice of fixing the engine or airframe system matrices will be presented. If neither is to be fixed (the usual choice) enter 0.

Various options for fixing the D-submatrices are presented. If none are to be fixed enter 0 here.

During the linesearch procedure, the user will see function and derivative values which indicate the progress of the search for a reduction in the function value.

After every major iteration of the program (starting from the second), a MATRIX  $\chi$  graph is generated showing the costs (total, partitioning and tracking) for the last twenty iterations.

• When the run ends, the output of START is the final optimized state-space representations for the subcontrollers SKA\_OPT and SKE\_OPT.

The data file INTER.DAT stores the history of the costs, the most recent parameter vector, approximation to the inverse Hessian matrix and gradient. This information can be used to see the progress of the algorithm, and to restart the program either after a successful termination or after user interruption.

There is a routine called RESTART which is available for restarting from a system crash or an intentional interruption. It uses the data stored in INTER.DAT and CONST.DAT and allows the user to define new values of the tracking cost weight  $\lambda$ , the stopping criteria STOP, and/or a new inverse Hessian approximation.

- The inputs and outputs of the RESTART routine are similar to those for START.

  There are three alternatives:
  - 1) enter [SKA\_OPT,SKE\_OPT]=RESTART(LAMBDANEW) if all that is changed is the LAMBDA weighting parameter or
  - 2) enter [SKA\_OPT,SKE\_OPT]=RESTART(LAMBDANEW, STOP) if a change is made in LAMBDA and/or STOP or
  - 3) or enter [SKA\_OPT,SKE\_OPT]=RESTART(LAMBDANEW, STOP, 1) if a restart with the identity matrix as the initial approximation to the inverse Hessian is desired. Note that you must enter values of LAMBDANEW and STOP even if they are the same as the previous LAMBDA and STOP.

The restarted program executes in the same way as before, with results stored in the file INTER.DAT and the output of the program being the optimized state-space representations SKA\_OPT and SKE\_OPT.

#### 6. Example of Controller Partitioning

#### STOL Example.

The controller partitioning software is first applied to a centralized flight/propulsion controller for a STOL aircraft as was described in reference [6]. This controller has the form  $\mathbf{u} = K(s)\mathbf{e}$  with the error vector  $\mathbf{e}$  consisting of errors,  $\mathbf{e} = [e_v, e_q, e_{N2}, e_{EPR}]^{\mathsf{T}}$ , in following velocity (v), pitch rate variable  $(q_v = q + 0.1\theta)$ , engine fan speed (N2) and engine pressure ratio (EPR) commands. The control input vector  $\mathbf{u}$  consists of rates of change of thrust vectoring angle, fuel flow, thrust reverser port area and nozzle throat area,  $\mathbf{u} = [\delta_{TV}, WF, A78, A8]^{\mathsf{T}}$ .  $\mathbf{u}$  consists of rates because integrators were appended to the control inputs during the process of centralized control design to achieve zero steady-state error for step commands. The partitioned airframe and engine controllers are desired to have inputs  $\mathbf{e}_a = [e_v, e_q]^{\mathsf{T}}$  and  $\mathbf{e}_e = [e_{N2}, e_{EPR}]^{\mathsf{T}}$  and outputs  $\mathbf{u}_a = [\delta_{TV}]$  and  $\mathbf{u}_e = [WF, A78, A8]^{\mathsf{T}}$  respectively. The interface variable  $\mathbf{z}_{ea}$  for this example is the single variable FEX, the axial thrust generated by the propulsion system. An initial controller partitioning was obtained using the procedure discussed in [5].

The numbers of airframe and propulsion subcontroller inputs are thus (MA=)2 and (ME=)2 while the subcontrollers have (KA=)1 and (KE=)3 outputs respectively. There is (PEA=)1 interface variable and no direct measurements are fed back from the integrated plant to the subcontrollers (LA and LE are absent from INIT.DAT since there are no measurements).

State-space matrices for the integrated plant, SP of order (NP=)13, the centralized controller, SC of order (NSC=)13 and initial partitioning, S\_KA of order (NS\_KA=)10 and S\_KE of order (NS\_KE=)7 are listed in Appendix IV.

The optimization is done over the frequency range  $\omega \in [0.1, 100]$  with 41 frequency points (FRQ=[0.1; 100; 41]). The frequency weighting

$$W_i(s) = G(s)(I + K(s)G(s))^{-1}$$

is used to achieve good performance matching as well as stability robustness for the equivalent controller. The state-space representation for this weighting, SWI of order (NWI=)26, is obtained from SP and SC. The tracking normalization, N\_TRACK=  $||T_{Cent}(j\omega)||_2^2$ , is used to scale the tracking cost. The state-space representation for  $T_{Cent}(s)$  is constructed as in expression (I.3) in Appendix I. Since LA and LE are absent, any blocks involving  $y_a$  or  $y_e$  as either inputs or outputs are omitted. Notice that since PEA=1,  $T_{Cent}$  is a 1× MA row vector. Neither output weighting (SWO) nor performance normalization (N\_PERF) were used.

All the necessary variables are stored in INIT.DAT.

The tracking weighting parameter LAMBDA is set to 0.05 and the stopping criteria vector is defined as

$$STOP = [1e-9;1e-9;1e-9;100;0.1].$$

After defining the UDF START by entering define 'START.MTX'

the program is started by entering [SKA,SKE]=START(LAMBDA,STOP).

Respond to the question concerning fixing the engine or airframe by entering '0' to fix neither as shown in Figure 6.1.

The '0' response to the question concerning the D- matrices as shown in Figure 6.1 will fix none of these submatrices. The program now begins.

ENTER 1 TO FIX AIRFRAME, 2 TO FIX ENGINE or 0 FOR NEITHER: 0

ENTER 1 TO FIX ALL Ds EXCEPT DEAA & DEEA, 2 TO INCLUDE DEEA, 3 FOR ONLY DEEA, or 0 for NONE: 0

Figure 6.1 Screen After Responding to Questions

During a major iteration, the user will see values of AL, FAL, FPAL, ALPR, FALPR, A, FA, FPA, B, FB and FPB displayed on the screen. These values result from calculations during the linesearch as described in Step 4 of Section 3. In particular [AL, ALPR] is the interval used in the "bracketing" phase of the linesearch and [A, B] is the interval used during "sectioning". The function values ( and directional derivatives) at the endpoints of these intervals are denoted by FAL, FALPR, FA, and FP (respectively FPAL, FPALPR, FPA and FPB). The user can follow the progress of the linesearch by viewing the values displayed on the screen as in Figure 6.2.

AL	=
	7.5865D-09
FAL	=
	4.9981D+01
FPAL	=
	6.7610D+08
ALPR	<del>=</del>
	0.
FALPR	=
	4.9737D+01
FPALPR	=
	-6.1169D+08

Α	=		
c	).		
FA	=		
	4.9737D+01		
FPA	=		
-	6.1169D+08		
В	=		
	7.5865D-09		
FB	=		
	4.9981D+01		
FPB	=		
	6.7610D+08		

DURING BRACKETING

DURING SECTIONING

Figure 6.2 Screen Display During the Linesearch

After the first linesearch succeeds the user will see a graphical display of the values of the total, performance and tracking costs for the previous major iterations (after the twentieth, only the last twenty are displayed). A typical screen is displayed in Figure 6.3.

After 100 iterations, SKA and SKE are returned. The convergence criteria were not all met, rather the program stopped because the maximum number of iterations was reached. Nonetheless, as will be seen by a posterioiri analysis, the resulting subcontrollers exhibited good performance and tracking properties. The total cost history is shown in Figure 6.4.

The performance of the initial controller partitioning is evaluated in comparison with that of the centralized controller by comparing closed-loop system response to step commands in the controlled variables z.

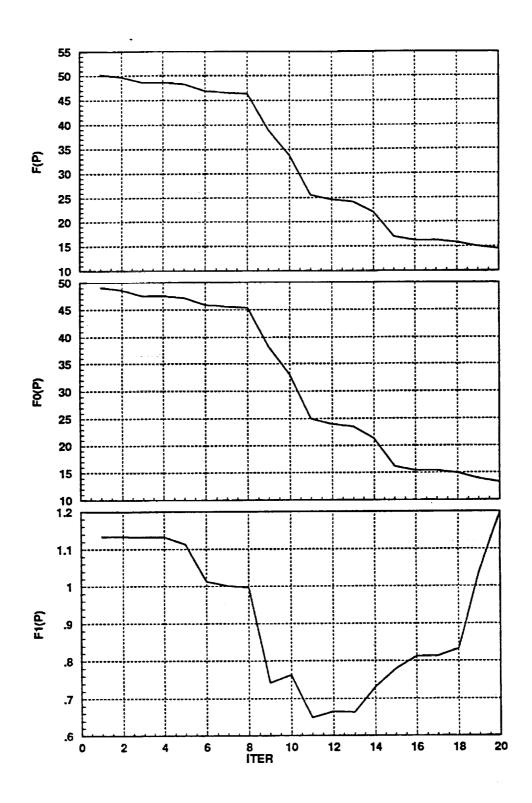


Figure 6.3 Screen Display Showing Total, Performance and Tracking Costs

The responses to  $q_{vc}$ ,  $N2_c$  and  $EPR_c$  with the initial as well as the optimized parti-

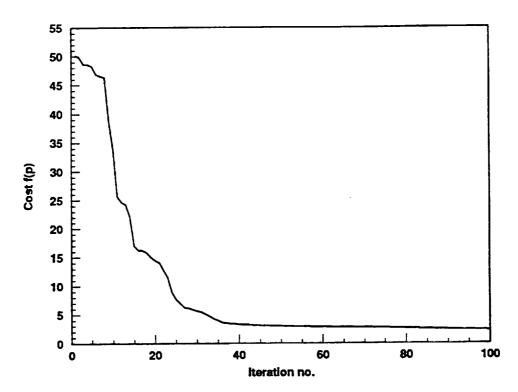


Figure 6.4 Cost History for Controller Partitioning Optimization

tioned subcontrollers were comparable to those with the centralized controller so they are not shown here. However, the responses to  $V_c$ , shown in Fig. 6.5, show considerable degradation in terms of increased coupling in the N2 and EPR responses with the initial partitioned subcontrollers. This deficiency was overcome by the optimized partitioned controllers as can be seen in Fig. 6.5. Note that all the quantities shown in Fig. 6.5 are normalized, using scalings discussed in [6], to allow a direct comparison of the various response magnitudes. In addition, the response of FEX (the interface variable) to  $V_c$  using partitioned subcontrollers was also comparable to that using the centralized controller as is seen in Fig. 6.6.

Since the performance with the optimized subcontrollers is found to be acceptable, an effort is made to reduce the orders of the subcontrollers. The engine subcontroller is reduced to 4<sup>th</sup> order by residualization of the three high frequency modes without any loss of performance. Through the use of internally balanced reduction techniques [7], the airframe subcontroller is reduced to 6<sup>th</sup> order (from the original 10<sup>th</sup> order) without excessive mismatch in the controller transfer matrix characteristics as is seen from the

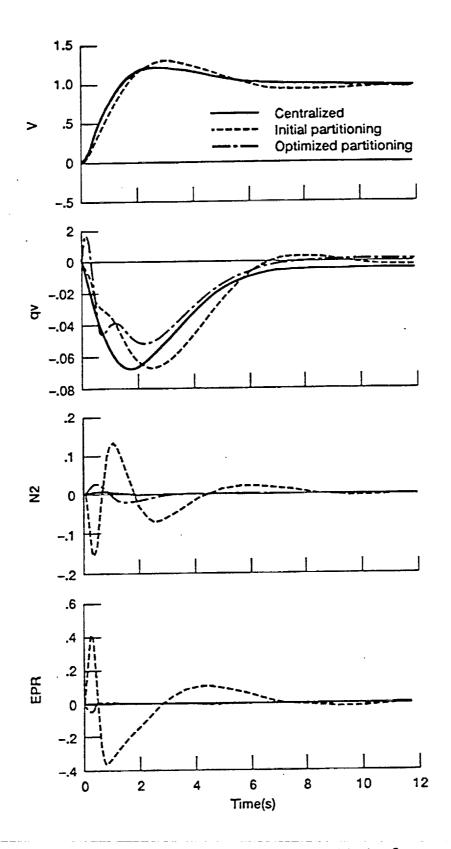


Figure 6.5 Closed-Loop System Response to Step Velocity Command for Centralized Controller, Initial Partitioning and Optimized Partitioning

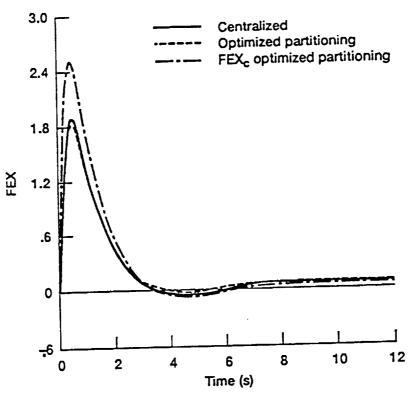


Figure 6.6  $\,FEX$  Response to Step Velocity Command for Centralized Controller and Optimized Partitioning

full and reduced order airframe controller singular values comparison in Fig. 6.7. This reduced order airframe subcontroller does, however, exhibit deterioration in closed-loop performance in the V and  $q_v$  response comparison plots for a step change in  $V_c$  as shown in Fig. 6.8.

The reduced order subcontroller state-space matrices are stored as S\_KA (order NS\_KA=6) and S\_KE (order NS\_KE=4) in INIT.DAT and the program is started by entering [SKA,SKE]=START(LAMBDA,STOP).

One should not use RESTART here since the CONST.DAT file will not contain data related to the reduced order subcontrollers. Moreover, since the engine subcontroller is acceptable, the optimization should take place over only the airframe subcontroller parameters. When requested to enter a value to fix a subcontroller, enter '2' to fix the engine subcontroller.

The program will execute as before, generating optimal SKA and SKE (fixed to the initial reduced order S\_KE). The response obtained with the optimized reduced order air-frame subcontroller for step  $V_c$  is shown in Fig. 6.8. Note that the optimized subcontroller

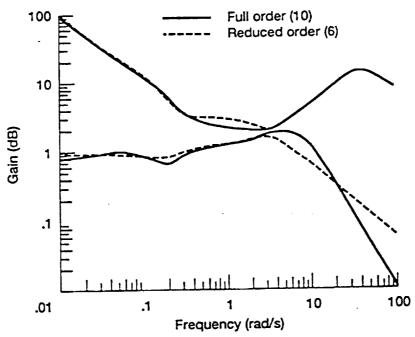


Figure 6.7 Singular Values of the Airframe Subcontroller for Optimized Partitioning with  $\lambda=0.05$  — Full (10) and Reduced (6) Orders

also provides improved tracking of the velocity command. The state-space matrix for the optimized reduced order airframe subcontroller is listed in Appendix IV.

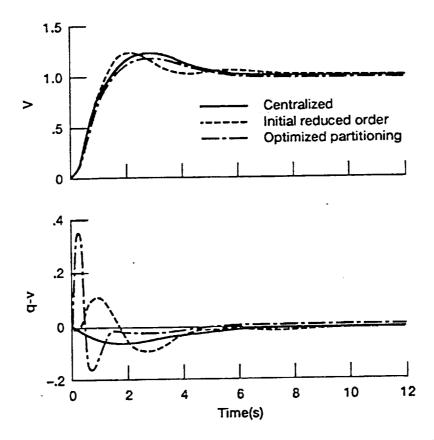


Figure 6.8 Closed-Loop System Response to Step Velocity Command for Centralized Controller and Reduced Order Partition Subcontrollers — Initial and Optimized

For a discussion of the application of this code to the design of a decentralized controller for a Short TakeOff and Vertical Landing (STOVL) aircraft, the user is referred to [8]. This application uses measurements from the integrated plant to the subcontrollers and includes more than one interface variable. In addition, several different weighting matrices are discussed in [8].

#### REFERENCES

- [1] Sanjay Garg, Peter J. Ouzts, Carl F. Lorenzo and Duane L. Mattern, IMPAC An integrated methodology for propulsion and airframe control, Proceedings of the 1991 American Control Conference, June, 1991, Boston, MA, 1 (1991), 747-754.
- [2] Dale F. Enns, Model Reduction for Control Systems Design, Ph.D. dissertation, Stanford University, 1984.
- [3] Phillip H. Schmidt, Sanjay Garg and Brian Holowecky, A parameter optimization approach to controller partitioning for integrated flight/propulsion control application, Proceedings of the 1992 Conference on Control Applications, September, 1992, Dayton, OH, 2 (1992), 972-979.
- [4] R. Fletcher, Practical Methods of Optimization, Wiley, New York, 1987.

- [5] Sanjay Garg, Controller Partitioning for Integrated Flight/Propulsion Control Implementation, presented at the 1992 American Control Conference, June, 1992, Chicago, IL.
- [6] Sanjay Garg, Duane L. Mattern and Randy E. Bullard, Integrated flight/propulsion control system design based on a centralized approach, AIAA Paper No. 89-3519, Proceedings of the AIAA Guidance, Navigation and Control Conference, Aug. 1989, Boston, MA.
- [7] B.C. Moore, Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction, IEEE Transactions on Automatic Control 26 (1981), 17-31.
- [8] Sanjay Garg and Phillip H. Schmidt, Application of Controller Partitioning Optimization Procedure to Integrated Flight/Propulsion Control Design for a STOVL Aircraft, AIAA Paper No. 93-3766, to appear in Proceedings of the AIAA Guidance, Navigation and Control Conference, Aug. 1993, Monterey, CA.
- [9] Uy-Loi Ly, Arthur E. Bryson and Robert H. Cannon, Design of low order compensators using parametric optimization, Automatica 21 (1985), 315-318.
- [10] Phillip H. Schmidt and Sanjay Garg, Decentralized Hierarchical Partitioning of Centralized Integrated Controllers, Proceedings of the 1991 American Control Conference, Boston, MA, June, 1991 1 (1991), 755-760.

#### APPENDIX I

#### DEVELOPMENT OF THE COST AND ITS GRADIENT

The Parameters. Parameters in the optimization process are certain entries in state-space realizations of  $K^a(s)$  and  $K^e(s)$  as defined in the formula (I.1). The notation  $\mathcal{M}_{oi}^s$  is used throughout to indicate the matrix  $\mathcal{M} \in \{A, B, C, D\}$  in the state-space realization of the system transfer matrix  $s \in \{c, p, a, e\}$  (c =centralized controller, p =plant, a =airframe subcontroller, e =engine subcontroller) with input i (respectively output o)  $\in \{p, a, e, ea_c, ea\}$  ( $ea_c$  =interface variable commands, ea =interface variables).

$$K^{a}(s): \begin{bmatrix} \mathbf{u}_{a} \\ \mathbf{z}_{eac} \end{bmatrix} = \begin{pmatrix} \begin{pmatrix} C_{aa}^{a} \\ C_{eac}^{a} \end{pmatrix} (sI - A^{a})^{-1} \begin{pmatrix} B_{aa}^{a} & B_{ay_{a}}^{a} \end{pmatrix} + \begin{pmatrix} D_{aa}^{a} & D_{ay_{a}}^{a} \\ D_{eac}^{a} & D_{eacy_{a}}^{a} \end{pmatrix} \end{pmatrix} \begin{bmatrix} \mathbf{e}_{a} \\ \mathbf{y}_{a} \end{bmatrix}$$

$$K^{e}(s): \quad \mathbf{u}_{e} = \begin{pmatrix} C_{ee}^{e} (sI - A^{e})^{-1} (B_{eea}^{e} & B_{ee}^{e} & B_{ey_{e}}^{e} \end{pmatrix} + (D_{eea}^{e} & D_{ee}^{e} & D_{ey_{e}}^{e} \end{pmatrix}) \begin{bmatrix} \mathbf{e}_{ea} \\ \mathbf{e}_{e} \\ \mathbf{y}_{e} \end{bmatrix}.$$

For the purposes of this software description, the corresponding state space matrices are written as

$$SKA = \begin{bmatrix} A^a & B^a_{aa} & B^a_{ay_a} \\ C^a_{aa} & D^a_{aa} & D^a_{ay_a} \\ C^a_{aa} & D^a_{aa} & D^a_{ay_a} \end{bmatrix} \quad \text{and} \quad SKE = \begin{bmatrix} A^e & B^e_{eea} & B^e_{ee} & B^e_{ey_e} \\ C^e_{ee} & D^e_{eea} & D^e_{ee} & D^e_{ey_e} \end{bmatrix} \quad (I.1)$$

One consideration in choosing a parameterization is to introduce a "minimal" number of parameters in the optimization process. A real canonical form used in [9] served as the model for our parameterization. The subcontroller system dynamics matrices  $A^a$  and  $A^e$  are represented as block diagonal matrices with two-by-two real companion blocks of the form  $\begin{bmatrix} 0 & 1 \\ \alpha & \beta \end{bmatrix}$ . If the order of either  $A^a$  or  $A^e$  is odd, there is also one diagonal real entry corresponding to a real eigenvalue. In addition,  $\alpha$  and  $\beta$  are constrained to be negative in order to meet the requirement that subcontrollers be stable. It should be noted that this form for the A matrices does not allow for a Jordan block structure. However, since the matrices are obtained from a numerical process, it is improbable that the "optimal" solution would need such a special structure.

In addition the first columns of each of the subcontroller input matrices  $B^a_{aa}$  and  $B^e_{eea}$  are fixed at non-zero values determined as follows. State space representations for the initial partitioning are not required to be in the canonical form described above. For each of them, a similarity

transformation, T, is applied to the initial A matrix so that  $TAT^{-1}$  is in the proper form. If A is an  $n \times n$  matrix then there will be n degrees of freedom in the determination of T. Different T will yield the same canonical  $TAT^{-1}$  but different transformed TB and  $CT^{-1}$  matrices. This implies that there are actually n degrees of freedom in the determination of TB and  $CT^{-1}$ . We select a simple T which is nonsingular and compute TB and  $CT^{-1}$ . We remove the degrees of freedom by fixing the n entries in the first column of TB to their values or  $10^{-9}$  if the corresponding value is zero.

We are using the  $H_2$  norm of the weighted difference between the transfer matrices for the centralized controller and the equivalent heirarchically partitioned subcontrollers as will be described below. Since this difference must be strictly proper in order to apply this norm, it is reasonable for the D matrices for the centralized and partitioned controllers to be the same. Thus it may be desirable to fix the values of  $D_{aa}^a$ ,  $D_{ay_a}^a$ ,  $D_{ee}^e$ , and  $D_{ey_e}^e$  to values determined directly by the centralized controller. This is one of the options available in "fixing the D-parameters".

The parameters over which the optimization takes place are then the  $\alpha$  and  $\beta$  entries in the block canonical forms, the entries in all but the first columns of the matrices  $B_{aa}^a$  and  $B_{eea}^e$  and all the entries in the matrices  $\begin{bmatrix} C_{aa}^a \\ C_{eea}^a \end{bmatrix}$ ,  $C_{ee}^e$ ,  $D_{eaa}^a$  and  $D_{eea}^e$ . The parameter vector will be denoted as  $\mathbf{p} \in \mathbb{R}^N$  where  $N = n_a(k_a + m_a + l_a + p_{ea}) + n_e(k_e + m_e + l_e + p_{ea}) + p_{ea}(k_a + m_e)$ ; n, m, k, and l refer respectively to the order, number of error inputs, number of outputs, and number of direct measurements for a subcontroller and  $p_{ea}$  refers to the number of interface variables. The number of parameters depends not only on the total numbers of controller inputs and outputs and interface variables which are fixed but also on the orders of the subcontrollers,  $n_a$  and  $n_e$ . There is thus a double incentive for keeping these orders low; not only to reduce the complexities of the subcontrollers but also to accelerate the optimization algorithm whose performance depends on the total number of parameters.

The Cost Function. The cost function is the sum of  $f_0(\mathbf{p})$  as in (2.2) and  $f_1(\mathbf{p})$  as in (2.3). These involve the transfer matrices K(s),  $\widetilde{K}(s)$ ,  $T_{\text{cent}}(s)$ , and  $\widehat{T}(s)$  which are described below.

The parameters in the cost function are the entries in SKA and SKE as described in the previous section. These parameters along with some fixed transfer matrices are used in the determination of the transfer matrices needed in the cost function.

State-space representations for the centralized controller transfer matrix, K(s), and the plant transfer matrix,  $\widehat{G(s)} = \begin{bmatrix} G(s) \\ G_{ex}(s) \end{bmatrix}$ , are given. Those for  $\widetilde{K}(s)$ ,  $T_{\text{cent}}(s)$  and  $\widehat{T}(s)$  (shown in formulas (I.2), (I.3) and (I.4) below) are constructed from the state-space representations of K(s),  $K^a(s)$ ,  $K^e(s)$ ,  $G_{ex}(s)$  and G(s).

The transfer matrix  $\tilde{K}(s)$  which enters into the performance cost term  $f_0(\mathbf{p})$  depends on  $K^a(s)$  and  $K^e(s)$ , and on the transfer submatrix of the plant from control inputs (u) to interface variables  $(\mathbf{z}_{\mathbf{e}a}), G_{\mathbf{e}a}: \mathbf{z}_{\mathbf{e}a} = C^p_{\mathbf{e}ap}(sI - A^p)^{-1} \begin{bmatrix} B^p_{pa} & B^p_{pe} \end{bmatrix} \begin{bmatrix} \mathbf{u}_a \\ \mathbf{u}_e \end{bmatrix}$ . The block diagram in Fig. I.1 shows the specific interconnections accounted for in this transfer matrix. Note that  $\tilde{K}(s)$  has the same inputs and outputs as K(s) as described in (2.1).

A state-space realization for the equivalent partitioned controller  $\widetilde{K}(s) = \widetilde{C}(sI - \widetilde{A})^{-1}\widetilde{B} + \widetilde{D}$  was shown in  $[\ 10\ ]$  to be  $\widetilde{S} = \begin{bmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{C} & \widetilde{D} \end{bmatrix}$  where

$$\widetilde{A} = \begin{bmatrix}
A^{a} & 0 & 0 \\
B^{e}_{eea}C^{a}_{eac} & A^{e} & -B^{e}_{eea}C^{p}_{eap} \\
(B^{p}_{pa}C^{a}_{aa} + B^{p}_{pe}D^{e}_{eea}C^{a}_{eac}a) & B^{p}_{pe}C^{e}_{ee} & (A^{p} - B^{p}_{pe}D^{e}_{eea}C^{p}_{eap})
\end{bmatrix}$$

$$\widetilde{B} = \begin{bmatrix}
B^{a}_{aa} & 0 & B^{a}_{ay_{a}} & 0 \\
B^{e}_{eea}D^{a}_{eac}a & B^{e}_{ee} & B^{e}_{eea}D^{a}_{eacy_{a}} & B^{e}_{ey_{e}} \\
(B^{p}_{pa}D^{a}_{aa} + B^{p}_{pe}D^{e}_{eea}D^{a}_{eaca}) & B^{p}_{pe}D^{e}_{ee} & (B^{p}_{pe}D^{e}_{eea}D^{a}_{eacy_{a}} + B^{p}_{pa}D^{a}_{ay_{a}}) & B^{p}_{pe}D^{e}_{ey_{e}}
\end{bmatrix}$$

$$\widetilde{C} = \begin{bmatrix}
C^{a}_{aa} & 0 & 0 \\
D^{e}_{eea}C^{a}_{eaca} & C^{e}_{ee} & -D^{e}_{eea}C^{p}_{eap} \\
D^{e}_{eea}D^{a}_{eaca} & D^{e}_{ee} & D^{e}_{eea}D^{a}_{eacy_{e}} & D^{e}_{ey_{e}}
\end{bmatrix}$$

$$\widetilde{D} = \begin{bmatrix}
D^{a}_{aa} & 0 & D^{a}_{ay_{a}} & 0 \\
D^{e}_{eea}D^{a}_{eaca} & D^{e}_{ee} & D^{e}_{eea}D^{a}_{eacy_{e}} & D^{e}_{ey_{e}}
\end{bmatrix}$$
(I.2)

The calculation of the tracking cost in (2.3) requires two transfer matrices,  $T_{\rm cent}(s)$  and  $\widehat{T}(s)$ . The norms of the rows of their difference measures the differences in response of the various possible interface variables to airframe commands. Figure I.2 is the block diagram for  $T_{\rm cent}(s) = C^{\rm cent}(sI - A^{\rm cent})^{-1}B^{\rm cent} + D^{\rm cent}$ , the closed loop transfer matrix from airframe commands,  $\mathbf{z}_{a_c}$ , to interface variable quantities,  $\mathbf{z}_{ea_c}$ , produced by the engine using the centralized controller.

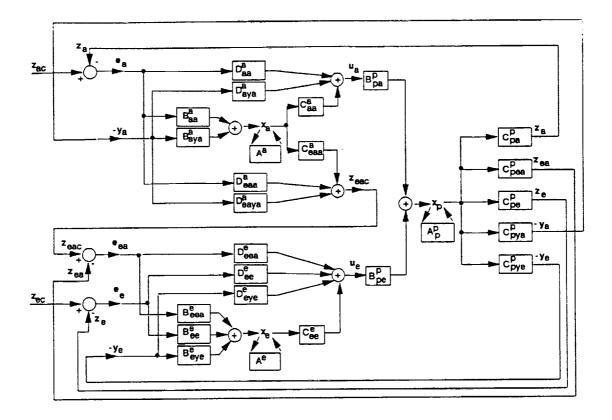


Figure 1.1 Partitioned Controller

Note that in Figures I.1, I.2 and I.3 a block of the form

As is easily seen from Figure I.2 the state-space representation for the centralized controller transfer matrix  $T_{\text{cent}}(s)$  can be written in terms of the submatrices in the state-space representations for K(s) and G(s),

$$A^{\text{cent}} = \begin{bmatrix} A^{c} & (-B_{ca}^{c}C_{ap}^{p} - B_{ce}^{c}C_{ep}^{p} + B_{cy_{a}}^{c}C_{y_{a}p}^{p} + B_{cy_{e}}^{c}C_{y_{e}p}^{p}) \\ (B_{pa}^{p}C_{ac}^{c} + B_{pe}^{p}C_{ec}^{c}) & A^{p} + B_{pa}^{p}(-D_{aa}^{c}C_{ap}^{p} + D_{ay_{e}}^{c}C_{y_{a}p}^{p} + D_{ae}^{c}C_{ep}^{p} + D_{ay_{e}}^{c}C_{y_{e}p}^{p}) \\ + B_{pe}^{p}(-D_{ee}^{c}C_{ep}^{p} + D_{ey_{e}}^{c}C_{y_{a}p}^{p} + D_{ee}^{c}C_{ep}^{p} + D_{ey_{e}}^{c}C_{y_{e}p}^{p}) \end{bmatrix}$$

$$B^{\text{cent}} = \begin{bmatrix} B_{ca}^{c} \\ B_{pa}^{p}D_{aa}^{c} \end{bmatrix} \qquad C^{\text{cent}} = \begin{bmatrix} 0 C_{eap}^{p} \end{bmatrix} \qquad D^{\text{cent}} = 0.$$
(I.3)

Figure I.3 is the block diagram for  $\widehat{T}(s) = \widehat{C}(sI - \widehat{A})^{-1}\widehat{B} + \widehat{D}$ , the closed loop transfer matrix from airframe commands,  $\mathbf{z}_{a_c}$ , to interface variable quantities produced by the airframe subcontroller,  $\mathbf{z}_{ea_c}$  using the partitioned controller.

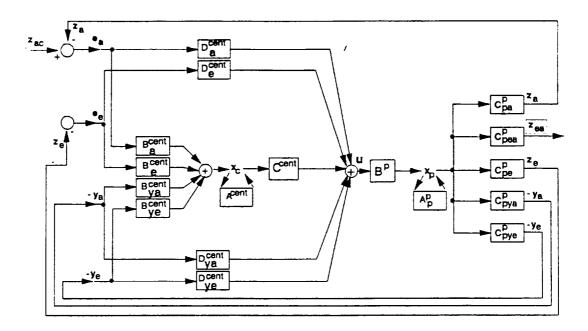


Figure 1.2  $\mathbf{z}_{\mathbf{z}_{\mathbf{z}}}$  response to  $\mathbf{z}_{\alpha_{\mathbf{z}}}$  using the centralized controller

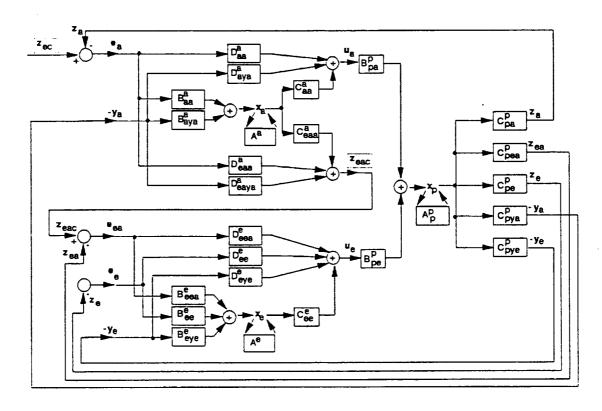


Figure 1.3  $\mathbf{z}_{ea_e}$  response to  $\mathbf{z}_{a_e}$  using the partitioned controller

The state space matrices for  $\widehat{T}(s) = \widehat{C}(sI - \widehat{A})^{-1}\widehat{B} + \widehat{D}$  can be written in terms of the state-space representations for  $K^a(s)$ ,  $K^e(s)$  and G(s) as

$$\widehat{T} = \begin{bmatrix} \widehat{A} & \widehat{B} \\ \widehat{C} & \widehat{D} \end{bmatrix} \quad \text{where}$$

$$\widehat{A} = \begin{bmatrix} A^{a} & 0 & -B^{a}_{aa}C^{p}_{ap} + B^{a}_{ay_{a}}C^{p}_{y_{a}p} \\ B^{e}_{eea}C^{a}_{ea_{c}a} & A^{e} & \widehat{A}_{ep} \\ B^{p}_{pa}C^{a}_{aa} + B^{p}_{pe}D^{e}_{eea}C^{a}_{ea_{c}a} & B^{p}_{pe}C^{e}_{ee} & \widehat{A}_{pp} \end{bmatrix}$$

$$\widehat{B} = \begin{bmatrix} B^{a}_{aa} \\ B^{e}_{eea}D^{a}_{ea_{c}a} \\ B^{p}_{pe}D^{a}_{aa} + B^{p}_{pe}D^{e}_{eea}D^{e}_{ea_{c}a} \end{bmatrix}$$

$$\widehat{C} = \begin{bmatrix} C^{a}_{ea_{c}a} & 0 & -D^{a}_{ea_{c}a}C^{p}_{p} + D^{a}_{ea_{c}y_{a}}C^{p}_{y_{a}p} \end{bmatrix}$$

$$\widehat{D} = \begin{bmatrix} D^{a}_{ea_{c}a} \end{bmatrix}$$

(I.4)

where  $\widehat{A}_{ep} = (-B_{ee}^e C_{ep}^p - C_{eea}^e C_{eap}^p - B_{eea}^e D_{ea_ca}^a C_{ap}^p + B_{eea}^e D_{ea_cy_a}^a C_{y_ap}^p + B_{ey_c}^e C_{y_ep}^p)$  and  $\widehat{A}_{pp} = (A^p - B_{pa}^p D_{aa}^a C_{ap}^p - B_{pe}^p D_{eea}^e D_{ea_ca}^a C_{ap}^p - B_{pe}^p D_{eea}^e C_{eap}^p - B_{pe}^p D_{ee}^e C_{ep}^p + B_{pa}^p D_{ay_a}^a C_{y_ap}^p + B_{pe}^p D_{eea}^e D_{ea_cy_a}^a C_{y_ap}^p + B_{pe}^p D_{eea}^e D_{ea_cy_a}^a C_{y_ap}^p + B_{pe}^p D_{eea}^e C_{y_ap}^p).$ 

It may be required that the engine subsystem be strictly proper, a condition which would be violated if  $D_{ee}^e$  is nonzero. As a result of the optimization process,  $D_{ee}^e$  may become large. This possibility is removed by "fixing  $D_{ee}^e = 0$ " when the option is presented while running the code.

The total cost is evaluated for a particular parameter vector  $\mathbf{p}$  (corresponding to particular SKA and SKE) by applying Simpson's Rule for numerical integration to  $f_0(\mathbf{p}) + f_1(\mathbf{p})$  over a user defined logarithmically spaced frequency interval  $[\omega_1, \omega_2]$ . The expressions given above for the state-space representations of K(s),  $\tilde{K}(s)$ ,  $T_{\text{cent}}(s)$  and  $\hat{T}(s)$  are used for calculating  $f_0(\mathbf{p})$  and  $f_1(\mathbf{p})$  according to formulas (2.2) and (2.3).

### The Gradient of the Cost.

The performance cost  $f_0(\mathbf{p})$  was defined in (2.2) as

$$f_0(\mathbf{p}) = \int_0^\infty \frac{1}{N_{\text{PERF}}} \operatorname{tr} \left[ \left( W_o(K - \widetilde{K(\mathbf{p})}) W_i \right)^* \left( W_o(K - \widetilde{K(\mathbf{p})}) W_i \right) \right] d\omega$$

where the explicit dependence of the integrand on  $\omega$  is suppressed for convenience.

Only  $\widetilde{K(\mathbf{p})}$  depends on  $\mathbf{p}$  so the derivative of  $f_0(\mathbf{p})$  with respect to a parameter  $\rho$  of  $\mathbf{p}$  is

$$\begin{split} \frac{\partial f_{0}(\mathbf{p})}{\partial \rho} &= \int\limits_{0}^{\infty} \frac{1}{N_{\mathrm{PERF}}} \mathrm{tr} \Big[ \frac{\partial}{\partial \rho} \left( W_{o}(K - \widetilde{K(\mathbf{p})}) W_{i} \right)^{*} \left( W_{o}(K - \widetilde{K(\mathbf{p})}) W_{i} \right) \\ &+ \left( W_{o}(K - \widetilde{K(\mathbf{p})}) W_{i} \right)^{*} \frac{\partial}{\partial \rho} \left( W_{o}(K - \widetilde{K(\mathbf{p})}) W_{i} \right) \Big] \mathrm{d}\omega \\ &= -2 \, \mathrm{Re} \int\limits_{0}^{\infty} \frac{1}{N_{\mathrm{PERF}}} \mathrm{tr} \left[ \left( W_{o}(K - \widetilde{K(\mathbf{p})}) W_{i} \right)^{*} W_{o} \frac{\partial \widetilde{K(\mathbf{p})}}{\partial \rho} W_{i} \right] \mathrm{d}\omega \end{split}$$

Since  $\widetilde{K(\mathbf{p})}(s) = \widetilde{C}(sI - \widetilde{A})^{-1}\widetilde{B} + \widetilde{D}$ , the product rule for differentiation implies that

$$\frac{\partial \widetilde{K(\mathbf{p})}(s)}{\partial \rho} = \frac{\partial \widetilde{C}}{\partial \rho} (sI - \widetilde{A})^{-1} \widetilde{B} + \widetilde{C}(sI - \widetilde{A})^{-1} \frac{\partial \widetilde{A}}{\partial \rho} (sI - \widetilde{A})^{-1} \widetilde{B} + \widetilde{C}(sI - \widetilde{A})^{-1} \frac{\partial \widetilde{B}}{\partial \rho} + \frac{\partial \widetilde{D}}{\partial \rho}$$

$$= \left[ \widetilde{C}(sI - \widetilde{A})^{-1} \quad I \right] \frac{\partial}{\partial \rho} \left[ \widetilde{A} \quad \widetilde{B} \atop \widetilde{C} \quad \widetilde{D} \right] \left[ (sI - \widetilde{A})^{-1} \widetilde{B} \right]$$

Thus,

$$\frac{\partial f_0(\mathbf{p})}{\partial \rho} = -2 \operatorname{Re} \int_0^\infty \frac{1}{N_{\text{PERF}}} \operatorname{tr} \left[ \left( W_o(K - \widetilde{K(\mathbf{p})}) W_i \right)^* \right]$$

$$W_o \left[ \widetilde{C} (j\omega I - \widetilde{A})^{-1} \quad I \right] \frac{\partial}{\partial \rho} \left[ \widetilde{\widetilde{C}} \quad \widetilde{\widetilde{D}} \right] \left[ (j\omega I - \widetilde{A})^{-1} \widetilde{\widetilde{B}} \right] W_i \right] d\omega$$

Using abbreviations for the system matrix given in (I.2) well as for the terms on the left and right sides of the partial,

$$\begin{split} \widetilde{S} &= \begin{bmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{C} & \widetilde{D} \end{bmatrix} \\ L(\omega) &= \frac{1}{N_{\text{PERF}}} \left( W_o(K - \widetilde{K(\mathbf{p})}) W_i \right)^* W_o \left[ \widetilde{C} (j\omega I - \widetilde{A})^{-1} \quad I \right] \\ R(\omega) &= \begin{bmatrix} (j\omega I - \widetilde{A})^{-1} \widetilde{B} \\ I \end{bmatrix} W_i \end{split}$$

allows the derivative to be written as

$$\frac{\partial f_0(\mathbf{p})}{\partial \rho} = -2 \operatorname{Re} \int_0^\infty \operatorname{tr} \left[ L(\omega) \frac{\partial \widetilde{S}}{\partial \rho} R(\omega) \right] d\omega.$$

Any particular parameter  $\rho$  of p is some  $jk^{th}$  entry of some submatrix, denoted  $\mathcal{M}_{oi}$ , relating an input i to an output o of SKA (or SKE) as described in (I.1). Furthermore,  $\rho$  (as well as the entire submatrix  $\mathcal{M}_{oi}$ ) occurs in one or more blocks of  $\widetilde{S} = \begin{bmatrix} \widetilde{A} & \widetilde{B} \\ \widetilde{C} & \widetilde{D} \end{bmatrix}$  as was described in (I.2).

 $\widetilde{S}$  can be thought of as consisting of blocks aligned in "columns" corresponding to the "inputs"  $\mathbf{x}_a$ ,  $\mathbf{x}_e$ ,  $\mathbf{x}_p$ ,  $\mathbf{e}_a$ ,  $\mathbf{e}_e$ ,  $\mathbf{y}_a$ ,  $\mathbf{y}_e$  and in "rows" corresponding to the "outputs"  $\mathbf{x}_a$ ,  $\mathbf{x}_e$ ,  $\mathbf{x}_p$ ,  $\mathbf{u}_a$ ,  $\mathbf{u}_e$ . Each block of  $\widetilde{S}$  is denoted as  $\mathcal{B}_{OI}$  where I is one of the "inputs" above and O is one of the "outputs". Let  $E_O$  denote a column block matrix with the same number of rows as  $\widetilde{S}$ , the same number of columns as the dimension of the "output" O, with an identity matrix in the rows corresponding to "output" O, and with zeros elsewhere. Let  $E_I^T$  be a similar row block matrix corresponding to the "input" I,

$$E_{\mathbf{O}} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad E_{\mathbf{I}}^{T} = \begin{bmatrix} 0 & \cdots & 0 & I & 0 & \cdots & 0 \end{bmatrix}.$$

Using this notation,  $\tilde{S}$  can be written as

$$\widetilde{S} = \sum_{\mathbf{O}} \sum_{\mathbf{I}} E_{\mathbf{O}} \mathcal{B}_{\mathbf{O}\mathbf{I}} E_{\mathbf{I}}^{T}.$$

In the partial derivative of  $\tilde{S}$  with respect to the  $\rho = \mathcal{M}_{\text{oi}jk}$ , every block of  $\tilde{S}$  is zero except for the blocks containing  $\rho$ . Denote such a block as  $\mathcal{B}_{\text{OI}} = \mathcal{L}_{\text{Oo}} \mathcal{M}_{\text{oi}} \mathcal{R}_{\text{iI}}$ , where  $\mathcal{L}_{\text{Oo}}$  denotes the factor to the left of  $\mathcal{M}_{\text{oi}}$  (if one exists, otherwise an identity) and  $\mathcal{R}_{\text{iI}}$  denotes the factor to the right( similarly, an identity if  $\mathcal{M}_{\text{oi}}$  is the rightmost factor); define both  $\mathcal{L}_{\text{Oo}} = 0$  and  $\mathcal{R}_{\text{iI}} = 0$  if  $\mathcal{B}_{\text{OI}}$  does not contain  $\mathcal{M}_{\text{oi}}$ . The partial of  $\tilde{S}$  with respect to  $\rho$  thus contains a term of the form  $\mathcal{L}_{\text{Oo}}\mathbf{e}_{j}\mathbf{e}_{k}{}^{T}\mathcal{R}_{\text{iI}}$  in the same place as the block  $\mathcal{B}_{\text{OI}}$ . These contributions can be written as

$$\frac{\partial \tilde{S}}{\partial \rho} = \sum_{\mathbf{O}} \sum_{\mathbf{I}} E_{\mathbf{O}} \mathcal{L}_{\mathbf{Oo}} \mathbf{e}_{j} \mathbf{e}_{k}^{T} \mathcal{R}_{i\mathbf{I}} E_{\mathbf{I}}^{T}$$

Thus the partial of the partitioning cost can be expressed as

$$\frac{\partial f_0(\mathbf{p})}{\partial \rho} = -2 \operatorname{Re} \int_0^\infty \operatorname{tr} \left[ L(\omega) \left( \sum_{\mathbf{O}} \sum_{\mathbf{I}} E_{\mathbf{O}} \mathcal{L}_{\mathbf{Oo}} \mathbf{e}_j \mathbf{e}_k^T \mathcal{R}_{i\mathbf{I}} E_{\mathbf{I}}^T \right) R(\omega) \right] d\omega.$$

Note that  $L(\omega)E_{O}\mathcal{L}_{Oo}e_{j}$  is a column vector and  $\mathbf{e}_{k}^{T}\mathcal{R}_{i\mathbf{I}}E_{\mathbf{I}}^{T}R(\omega)$  is a row vector. Furthermore, the trace of a column times a row is the dot product of the row and the column. Therefore,

$$\frac{\partial f_0(\mathbf{p})}{\partial \rho} = -2 \operatorname{Re} \int_0^\infty \sum_{\mathbf{Q}} \sum_{\mathbf{I}} \left[ (\mathbf{e}_k^T \mathcal{R}_{i\mathbf{I}} E_{\mathbf{I}}^T R(\omega)) (L(\omega) E_{\mathbf{Q}} \mathcal{L}_{\mathbf{Q} \circ} \mathbf{e}_j) \right] d\omega.$$

To generate the partials with respect to all the parameters in  $\mathcal{M}_{oi}$  simultaneously, let j and k vary over the row and column indicies respectively of the submatrix. Notice that varying the row index j selects the  $j^{th}$  column of the product  $L(\omega)E_{O}\mathcal{L}_{Oo}$ , whereas varying the column index k selects the  $k^{th}$  row of  $\mathcal{R}_{iI}E_{I}^{T}R(\omega)$ .

A matrix containing the partial derivatives with respect to the entries of  $\mathcal{M}_{oi}$  located in each entry's proper spot is thus obtained by replacing  $\mathbf{e}_j$  and  $\mathbf{e}_k$  by identity matrices and transposing the result.

$$\frac{\partial f_0(\mathbf{p})}{\partial \mathcal{M}_{oi}} = -2 \operatorname{Re} \int_0^\infty \sum_{\mathbf{O}} \sum_{\mathbf{I}} \left[ (\mathcal{R}_{i\mathbf{I}} E_{\mathbf{I}}^T R(\omega)) (L(\omega) E_{\mathbf{O}} \mathcal{L}_{\mathbf{Oo}}) \right]^T d\omega$$

Note that only the terms in  $R(\omega)L(\omega)$  depend on  $\omega$  and thus the integration can be rewritten to yield

$$\frac{\partial f_0(\mathbf{p})}{\partial \mathcal{M}_{oi}} = -2 \operatorname{Re} \left[ \sum_{\mathbf{O}} \sum_{\mathbf{I}} \mathcal{R}_{i\mathbf{I}} E_{\mathbf{I}}^T \left[ \int_0^{\infty} [R(\omega) L(\omega)] d\omega \right] E_{\mathbf{O}} \mathcal{L}_{\mathbf{Oo}} \right]^T$$

Finally, build two matrices denoted  $\frac{\partial f_0(\mathbf{p})}{\partial SKA}$  and  $\frac{\partial f_0(\mathbf{p})}{\partial SKE}$  of the shapes of SKA and SKE (described in (I.1)) respectively containing the partial derivatives of  $f_0(\mathbf{p})$  with respect to the parameters in SKA (respectively SKE) in the same positions as the corresponding parameters would occur. This is done so that in the software the gradient vector can be produced from  $\frac{\partial f_0(\mathbf{p})}{\partial SKA}$  and  $\frac{\partial f_0(\mathbf{p})}{\partial SKE}$  by a call of the function LONGCOL (the same function which produces  $\mathbf{p}$  from SKA and SKE).

To build these matrices, define the "row" and "column" block matrices  $E^a{}_i{}^T$  and  $E^a{}_o$  relative to the "inputs"  $\mathbf{i} \in \{\mathbf{x}_a, \mathbf{e}_a, \mathbf{y}_a\}$  and "outputs"  $\mathbf{o} \in \{\mathbf{x}_e, \mathbf{u}_a, \mathbf{z}_{\mathbf{e}a_e}\}$  for the airframe controller state-space system matrix SKA. As before  $E^a{}_o$  is a block column matrix with as many rows as SKA, with an identity matrix of size equal to the dimension of the output  $\mathbf{o}$  in the rows corresponding to  $\mathbf{o}$  and with zeros in the remaining rows. The matrices column block matrix  $E^e{}_o$  and the row block matrices  $E^a{}_i{}^T$ , and  $E^e{}_i{}^T$  are similarly defined. If we again denote the submatrices of SKA

as  $\mathcal{M}^a_{oi}$  and those of SKE as  $\mathcal{M}^e_{oi}$ , we can write

$$\mathrm{SKA} = \sum_{\mathbf{o}} \sum_{\mathbf{i}} E^{a}{}_{\mathbf{o}} \mathcal{M}^{a}{}_{\mathbf{o}\mathbf{i}} E^{a}{}_{\mathbf{i}}{}^{T} \quad \text{and} \quad \mathrm{SKE} = \sum_{\mathbf{o}} \sum_{\mathbf{i}} E^{e}{}_{\mathbf{o}} \mathcal{M}^{e}{}_{\mathbf{o}\mathbf{i}} E^{e}{}_{\mathbf{i}}{}^{T}.$$

Notice that, as with  $\tilde{S}$ , the column and row block matrices merely position each  $\mathcal{M}_{oi}$  properly. Thus, replacing  $\mathcal{M}_{oi}$  by the corresponding block  $\frac{\partial f_0(\mathbf{p})}{\partial \mathcal{M}_{oi}}$  in these formulas gives the desired results

$$\frac{\partial f_0(\mathbf{p})}{\partial SKA} = \sum_{\mathbf{o}} \sum_{\mathbf{i}} E^{a_{\mathbf{o}}} \frac{\partial f_0(\mathbf{p})}{\partial \mathcal{M}^{a_{\mathbf{o}i}}} E^{a_{\mathbf{i}}}^T$$

and

$$\frac{\partial f_0(\mathbf{p})}{\partial SKE} = \sum_{\mathbf{p}} \sum_{\mathbf{i}} E^{e_{\mathbf{p}}} \frac{\partial f_0(\mathbf{p})}{\partial \mathcal{M}^{e_{\mathbf{p}i}}} E^{e_{\mathbf{i}}}^{T}$$

Using the expression from above for  $\frac{\partial f_0(\mathbf{p})}{\partial \mathcal{M}^a_{oi}}$  gives

$$\frac{\partial f_0(\mathbf{p})}{\partial SKA} = -2 \operatorname{Re} \sum_{\mathbf{o}} \sum_{\mathbf{i}} E^{a}_{\mathbf{o}} \left[ \sum_{\mathbf{O}} \sum_{\mathbf{I}} \mathcal{R}_{\mathbf{i}\mathbf{I}} E_{\mathbf{I}}^{T} \left[ \int_{0}^{\infty} [R(\omega)L(\omega)] d\omega \right] E_{\mathbf{O}} \mathcal{L}_{\mathbf{Oo}} \right]^{T} E^{a}_{\mathbf{i}}^{T}$$

$$= -2 \operatorname{Re} \left[ \sum_{\mathbf{o}} \sum_{\mathbf{i}} \sum_{\mathbf{O}} \sum_{\mathbf{I}} E^{a}_{\mathbf{i}} \mathcal{R}_{\mathbf{i}\mathbf{I}} E_{\mathbf{I}}^{T} \left[ \int_{0}^{\infty} [R(\omega)L(\omega)] d\omega \right] E_{\mathbf{O}} \mathcal{L}_{\mathbf{Oo}} E^{a}_{\mathbf{o}}^{T} \right]^{T}.$$

The terms of the form  $E^a{}_i\mathcal{R}_{iI}E_I{}^T$  are independent of "outputs" o and O whereas those of the form  $E_0\mathcal{L}_{Oo}E^a{}_o{}^T$  are independent of "inputs" i and I. Therefore the sums can be rearranged

$$\frac{\partial f_0(\mathbf{p})}{\partial SKA} = -2 \operatorname{Re} \left[ \left( \sum_{\mathbf{i}} \sum_{\mathbf{I}} E^{a_{\mathbf{i}}} \mathcal{R}_{\mathbf{i}\mathbf{I}} E_{\mathbf{I}}^{T} \right) \left( \int_{\mathbf{0}}^{\infty} [R(\omega) L(\omega)] d\omega \right) \left( \sum_{\mathbf{o}} \sum_{\mathbf{O}} E_{\mathbf{O}} \mathcal{L}_{\mathbf{O}\mathbf{o}} E^{a_{\mathbf{o}}}^{T} \right) \right]^{T}.$$

The sum  $\sum_{i} \sum_{I} E^{a}_{i} \mathcal{R}_{iI} E_{I}^{T}$  represents a matrix with "inputs" I, the "inputs" of  $\widetilde{S}$ , and with "outputs" i, the "inputs" for SKA. This matrix has the submatrix  $\mathcal{R}_{iI}$  as the block in the rows corresponding to i and the columns corresponding to I; denote this matrix as  $\mathbf{R}^{a}$ . Similarly the matrix  $\mathbf{L}^{a} = \sum_{o} \sum_{O} E_{O} \mathcal{L}_{Oo} E^{a}_{o}^{T}$  contains the submatrix  $\mathcal{L}_{Oo}$  in its block with rows corresponding to the  $\widetilde{S}$  "output" O and with columns corresponding to SKA "outputs" o.

The following simple procedure can thus be used to determine  $\frac{\partial f_0(\mathbf{p})}{\partial SKA}$ :

1. For each block  $\mathcal{B}_{OI}$  of  $\widetilde{S}$  containing a submatrix  $\mathcal{M}^a_{oi}$  from SKA, determine the left and right factors (or identity)  $\mathcal{L}_{Oo}$  and  $\mathcal{R}_{iI}$  of  $\mathcal{M}^a_{oi}$ .

2. Enter  $\mathcal{L}_{\mathbf{Oo}}$  in the appropriate block of  $\mathbf{L}^a$  and  $\mathcal{R}_{iI}$  in the appropriate block of  $\mathbf{R}^a$ .

3. Calculate 
$$\int\limits_0^\infty \,R(\omega)L(\omega)\,\,\mathrm{d}\omega$$
 where 
$$L(\omega) = \frac{1}{N_{\mathrm{PERF}}} \left(W_o(K-\widetilde{K(\mathbf{p})})W_i\right)^*W_o\left[\,\widetilde{C}(j\omega I-\widetilde{A})^{-1}\quad I\,\right]$$
 
$$R(\omega) = \left[ \begin{array}{c} (j\omega I-\widetilde{A})^{-1}\widetilde{B} \\ I \end{array} \right]W_i$$

4. Form

$$\frac{\partial f_0(\mathbf{p})}{\partial SKA} = -2 \operatorname{Re} \left[ \mathbf{R}^a \left( \int_0^\infty [R(\omega)L(\omega)] d\omega \right) \mathbf{L}^a \right]^T.$$

By a similar procedure

$$\frac{\partial f_0(\mathbf{p})}{\partial SKE} = -2 \operatorname{Re} \left[ \mathbf{R}^{e} \left( \int_{0}^{\infty} [R(\omega)L(\omega)] d\omega \right) \mathbf{L}^{e} \right]^{T}$$

where L<sup>e</sup> and R<sup>e</sup> contain the left and right factors respectively of  $\mathcal{M}^{e}_{oi}$  terms appearing in  $\tilde{S}$ .

It is easy to use the representation of  $\widetilde{S}$  as given in (I.2) to calculate

$$\mathbf{R}^{a} = \begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{L}^{a} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & B_{eea}^{p} & 0 & B_{eea}^{p} \\ 0 & B_{pa}^{p} & B_{pe}^{p} D_{eea}^{e} \\ 0 & I & 0 \\ 0 & 0 & D_{eea}^{e} \end{bmatrix}$$

and

$$\mathbf{R}^{e} = \begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 \\ C^{e}_{eea} & 0 & -C^{p}_{eap} & D^{a}_{eac} & 0 & D^{a}_{eac} & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I \end{bmatrix} \quad \text{and} \quad \mathbf{L}^{e} = \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & B^{p}_{pe} \\ 0 & 0 \\ 0 & I \end{bmatrix}.$$

The Tracking Cost

The tracking cost was given in (2.3) as

$$f_1(\mathbf{p}) = \int_0^\infty \sum_i \frac{\lambda_i}{N_{\mathrm{TRACK}}^i(\omega)} \left( ||T_{\mathrm{cent}}^i - \widehat{T^i}(\mathbf{p})||_2 \right)^2 \mathrm{d}\omega$$

where the state space representations for  $T_{\text{cent}}(s)$  and  $\widehat{T}(s)$  are given in (I.3) and (I.4) respectively.

This function can be put into a form which is similar to  $f_0(\mathbf{p})$  so the same procedure for calculating the gradient applies. Each row of the difference is normalized before the sum of squares is calculated. Note that the multiplication by non-negative  $\lambda_i$  and division by non-negative  $N_{\text{TRACK}}^i(\omega)$  could also be absorbed into the normalized sum of squares by the use of their square roots. Multiplication (or division) of rows by factors can also be achieved by multiplication from the left by a diagonal matrix. The sum of squares of the row norms in the resulting product is the same as the sum of squares of all the entries or the  $H_2$  norm of the diagonal weighted difference. In this case

$$\begin{split} f_1(\mathbf{p}) &= \int\limits_0^\infty \mathrm{tr} \left[ \left( \mathrm{diag} \left( \sqrt{\frac{\lambda_i}{N_{\mathrm{TRACK}}^i(\omega)}} \; \right) \left( T_{\mathrm{cent}} - \widehat{T}(\mathbf{p}) \right) \right)^* \right. \\ & \left. \mathrm{diag} \left( \sqrt{\frac{\lambda_i}{N_{\mathrm{TRACK}}^i(\omega)}} \; \right) \left( T_{\mathrm{cent}} - \widehat{T}(\mathbf{p}) \right) \right] \mathrm{d}\omega. \end{split}$$

As before, denote

$$L_{t}(\omega) = \left(\operatorname{diag}\left(\sqrt{\frac{\lambda_{i}}{N_{\mathsf{TRACK}}^{i}(\omega)}}\right) \left(T_{\mathsf{cent}} - \widehat{T}(\mathbf{p})\right)\right)^{T}$$

$$\operatorname{diag}\left(\sqrt{\frac{\lambda_{i}}{N_{\mathsf{TRACK}}^{i}(\omega)}}\right) \left[\widehat{C}(j\omega I - \widehat{A})^{-1} \quad I\right]$$

$$R_{t}(\omega) = \begin{bmatrix} (j\omega I - \widehat{A})^{-1}\widehat{B} \\ I \end{bmatrix}$$

and apply the same procedure as earlier to write the partial derivatives as

$$\frac{\partial f_1(\mathbf{p})}{\partial SKA} = -2 \text{ Re} \left[ \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & -C_{ap}^p & I \\ 0 & 0 & C_{y_*p}^p & 0 \end{bmatrix} \int_0^\infty R_t(\omega) L_t(\omega) d\omega \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & B_{eea}^p \\ 0 & B_{pe}^p & B_{pe}^p D_{eea}^e \\ 0 & 0 & I \end{bmatrix} \right]^I$$

and

$$\frac{\partial f_1(\mathbf{p})}{\partial SKE} = -2 \operatorname{Re} \left[ \begin{bmatrix} 0 & I & 0 & 0 & 0 \\ C_{eea}^e & 0 & -C_{eap}^p - D_{eea}^e C_{eap}^p + D_{ea_c y_a}^a C_{y_a p}^p & 0 & D_{ea_c a}^a \\ 0 & 0 & -C_{ep}^p & 0 \end{bmatrix} \int_0^\infty R_t(\omega) L_t(\omega) \mathrm{d}\omega \begin{bmatrix} 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right]^T$$

This completes the discussion of the cost function and its gradient as implemented in this software. If the user wishes to compute cost functions involving the  $H_2$  norm as used here, the user must apply the procedure described above to determine the gradient.

### APPENDIX II

### SHORT DESCRIPTIONS OF USER-DEFINED FUNCTIONS

### START in START.MTX

 $[SKA\_OPT,SKA\_OPT] = START(LAMBDA, STOP)$ 

Initializes information from data files, puts subcontrollers into modified modal form, constructs or initializes some matrices required for the evaluation of the cost function. Calls COST to initialize the costs and gradients and then calls PARTITION to perform the optimization.

### INPUT

LAMBDA — weighting for the contribution of the tracking cost to the total cost.

STOP (optional) — vector of stopping conditions.

INIT.DAT — data file containing initial information.

### OUTPUT

SKA\_OPT — the state-space representation of the optimized airframe subcontroller.

SKE\_OPT — the state-space representation of the optimized engine subcontroller.

CONST.DAT — data file containing constants used by other UDFs.

PAR.DAT — used to store the stopping conditions in STOP.

### RESTART in RESTART.MTX

 $[SKA\_OPT,SKA\_OPT] = RESTART(LAMBDANEW, STOP, NEWH)$ 

Restarts the program using the data available in INTER.DAT and CONST.DAT. Calls COST to initialize the costs and gradients, and then calls PARTITION to optimize.

### INPUT

LAMBDANEW — a new value of LAMBDA may allow a different emphasis on the tracking cost relative to the total cost.

STOP (optional) — the vector of stopping conditions, same as in the START routine. It may be redefined, perhaps to allow more stringent conditions or more iterations.

NEWH (optional) — a flag whose presence indicates the desire for restart with identity inverse Hessian. If restart with the current Hessian matrix is desired, no value should be passed.

CONST.DAT — data file containing constants pertaining to the plant and global controller being partitioned.

INTER.DAT — the data file in which intermediate results from previous iterations are stored.

### OUTPUT

SKA\_OPT — the state-space representation of the optimized airframe subcontroller.

SKE\_OPT — the state-space representation of the optimized engine subcontroller.

PAR.DAT — the data file in which the stopping conditions STOP are saved.

INTER.DAT — the data file in which intermediate results are stored.

### PARTITION in PARTITIO.MTX

### PJ = PARTITION(STOP)

This is the main routine and does the optimization. It iterates till some convergence or stopping conditions are met. After each iteration a graph is plotted showing the change in costs. PARTITION implements the Broyden-Fletcher-Goldfarb-Shanno method of determining a search direction for minimization. It then calls the function INACCURATE to implement Fletcher's inaccurate line search in that direction. The function CONVERGE is called to see if convergence or stopping conditions are met; if not, the iteration in PARTITION is repeated.

### INPUT

STOP — the vector of stopping conditions, used to check convergence.

CONST.DAT — the data file containing the constants of the program.

INTER.DAT — the data file containing the intermediate results.

### OUTPUT

PI — the vector of parameters, after the ith optimization step.

INTER.DAT — the data file containing the intermediate results, where the results of the optimization are stored.

### INACCURATE in LINESRCH.MTX

[ALI, FAL, FP0, FP1, DFDPAL, DFP0, DFP1, FLAG]

= INACCURATE (X, FZ, DFDPZ, D, FMIN, DELTAF, LAMBDA, ALPHAMAX)

This function performs Fletcher's inaccurate line search as part of the unconstrained optimization performed by the function PARTITION. It calls the function COST to get the cost and gradient for the cost function evaluated at the  $i^{th}$  set of parameters. Effectively, INACCURATE seeks a point where a sufficient decrease in both the function value and directional derivative have occurred.

### INPUT

X — the current point (parameter vector), before the linesearch.

FZ — the total cost function evaluated at X.

DFDPZ — the gradient of the cost function evaluated at X.

D — the direction vector which the PARTITION function has chosen to perform the line search.

FMIN — the minimum value for the cost function. If the cost falls below this value, the function will terminate.

DELTAF — the estimated change in cost.

LAMBDA — the weighting for the tracking cost.

ALPHAMAX — maximum alpha, set by stability constraints.

### OUTPUT

ALI — the alpha value that yields sufficient reduction in  $F(X + alpha \times D)$  for the i<sup>th</sup> iteration.

FAL — the total cost function evaluated at  $X + AL \times D$ .

FP0 — the partitioning cost evaluated at  $X + AL \times D$ .

FP1 — the tracking cost evaluated at  $X + AL \times D$ .

DFDPAL — the gradient of the total cost function evaluated at  $X + AL \times D$ .

DFP0 — the gradient of the partitioning cost at  $X + AL \times D$ .

DFP1 — the gradient of the tracking cost at  $X + AL \times D$ .

FLAG — an output flag signalling the condition with which INACCURATE completed.

- 0 solution found, no problems.
- 1 solution found, objective value less than FMIN.
- 2 new point found, but backeting step converged to right endpoint without satisfying Armijo/Goldstein conditions.

### COST in COST.MTX

[DF, DF0, DF1, FP, FP0, FP1] = COST (P, LAMBDA)

Evaluates the cost and gradient at the current point using the cost function described in Appendix I. The state space representations for  $\tilde{K}(s)$  and  $\hat{T}(s)$  are constructed based on the current value of the parameter P. The contributions of these to the costs  $f_0$  and  $f_1$  are calculated at each frequency point in OMEGA and are added according to Simpson's Rule for numerical integration. Furthermore, the contributions to the gradient as described in Appendix I are also accumulated for each frequency in OMEGA.

### INPUT

P — the current point (parameter vector).

LAMBDA — the weighting for the tracking cost.

CONST.DAT — the data file containing the constants of the program.

### OUTPUT

DF, DF0, DF1 — the gradients of the total cost function, the partitioning cost function and the tracking cost function respectively.

FP, FP0, FP1 — the values of the total cost function, the partitioning cost function and the

tracking cost function respectively.

### CONVERGE in CONVERGE.MTX

ANSWER = CONVERGE (XJ1, FJ1, XJ, FJ, GRAD, EPS, DEL, ETA)

This function checks if the convergence condition are met and returns a flag ANSWER defined as

$$\text{ANSWER} = \left\{ \begin{array}{ll} 0 & \text{if MAX } |\text{X} \text{$\rfloor$1-X} \text{$\rfloor$}| > \text{EPS or } |\text{F} \text{$\rfloor$1-F} \text{$\rfloor$}| > \text{DEL} \\ 1 & \text{if MAX } |\text{X} \text{$\rfloor$1-X} \text{$\rfloor$}| < \text{EPS and } |\text{F} \text{$\rfloor$1-F} \text{$\rfloor$}| < \text{DEL but MAX } |\text{GRAD}| \geq \text{ETA} \\ 2 & \text{if MAX } |\text{X} \text{$\rfloor$1-X} \text{$\rfloor$}| < \text{EPS, } |\text{F} \text{$\rfloor$1-F} \text{$\rfloor$}| < \text{DEL and MAX } |\text{GRAD}| < \text{ETA} \\ \end{array} \right.$$

### Input

 $X_{11}$  — the value of the variable (parameter vector) at the  $(i+1)^{th}$  iteration.

F\_I1 — the value of the cost function at the (i+1)<sup>th</sup> iteration.

XJ — the value of the variable (parameter vector) at the  $i^{th}$  iteration.

FI — the value of the cost function at the  $i^{th}$  iteration.

GRAD — the gradient of the cost function at the  $(i+1)^{th}$  iteration.

EPS — the absolute tolerance for X differences.

DEL — the absolute tolerence for cost function differences.

ETA — the absolute tolerence for the gradient.

### OUTPUT

ANSWER — a flag indicating the state of convergence.

0 — no convergence.

1 — variable values converge, function values converge.

2 — variable values, function values, and gradient converge.

### MODL in MODL.MTX

[SM]=MODL(S,NS)

This function takes a system matrix and puts it in to the form where A has  $2 \times 2$  companion matrix blocks whose first rows are  $[0\ 1]$  and whose second rows are  $[a\ b]$ . The transformation matrix is

normalized by requiring that all nonzero entries in the first column of the B matrix remain fixed (zero entries are set at  $10^{-9}$ ).

INPUT

S — the system matrix.

NS — the order of the input matrix S.

OUTPUT

SM — the system matrix in modified modal form, still having order NS.

MAT in PARMAT.MTX

[SKA,SKE] = MAT(P)

Creates the system matrices for the airframe and the engine controllers from the parameter vector (the variable over which the optimization process is being performed). In case one of the subsystems is fixed, the corresponding initial system matrix is loaded from CONST.DAT and returned as SKA or SKE.

INPUT

P — the parameter vector.

CONST.DAT — the data file of constants.

OUTPUT

SKA — the system matrix for the airframe controller.

SKE — the system matrix for the engine controller.

LONGCOL in PARVEC.MTX

[P] = LONGCOL(SKA,SKE)

This function generates the parameter vector from the system matrices for the airframe and the engine. In case one of the subcontrollers is fixed, the parameter vector corresponds only to the

parameters in the other subcontroller.

### INPUT

SKA — the airframe controller.

SKE — the engine controller.

CONST.DAT — the data file of constants.

### OUTPUT

 ${\bf P}$  — the long column vector of the parameters.

### Z in ZERO.MTX

[ZER] = Z(NROW, NCOL)

Constructs a matrix of zeros of size NROW × NCOL.

### INPUT

NROW, NCOL — the row and column size of the desired zero matrix.

### OUTPUT

 ${\tt ZER}$  — the generated zero matrix.

APPENDIX III

CONTROLLER PARTITIONING CODE

converge.mtx ...nization algorithm a.
..psilon(j) for j-l..n where dlm(x)=.,-f(xi) | < delta
..unction values (costs) are converging
...d: |qrd(j)| < eta for j-l..n where dlm(x)=n x l
...d: |qrd(j)| < eta for j-l..n where dlm(x)=n x l
...d: |qrd(j)| < eta for j-l..n where dlm(x)=n x l
...d: |qrd(j)| < eta for j-l..n where dlm(x)=n x l
...d: |qrd(j)| < eta for j-l..n where dlm(x)=n x l
...d: |qrd(j)| < eta for j-l..n where dlm(x)=n x l
...d: |qrd(j)| < eta for j-l..n where dlm(x)=n x l
...d: |qrd(j)| < eta for the answer are:
...d: |qrd(j)| < for the answer are:
...d: |qrd(j)| < for the answer are:
...d: |qrd(j)| < eta for the answer are:
...d: |qrd(j)| < et PARTITION to test the convergence of the optimization algorithm according x i - (real,vec) value of variable at iteration (i).

f i and f ii - (real,scal) values of function at successive iterations g i and g ii - (real,vec) gradients at successive iterations epsilon - (real,vec) vector of the absolute tolerances defined Condition X: | xil(j) -xi(j) | < epsilon(j) for j-1..n where dlm(x) -n x 1 delta - [real, scal] absolute tolerance for the objective values. answer-converge (x\_11,f\_11,x\_1,f\_1,grd,epsilon,delta,eta,fx) eta - [real, scal] absolute tolerance for gradient Output:

fx - (real, vec) tracks which of the A matrices is at its stability bound answer - [int, scal] flag that will show whether or not the optimization procedure has converged according to the

user-defined criteria. (row, col) -size (grd); answer=0;

if abs(f il-f i) > delta then retf;
diff=abs(x\_il-x\_i); if diff(1)>epsilon(1), retf;... for 1=1:row; answer-1;

or i=1:row; ... if fx(i)-0, if grd(i)>eta, retf; ... for 1-1:row; end; end;

answer-2; retf;

Created: 01/24/90

Revised by Phil Schmidt and students Nader Kamrani and Brian Holawecky at The University of Akron with the support of NASA Lewis Research Center Programmer: Steven Ims under grant NAG-3-1146.

49

(Df, Df0, Df1, fp, fp0, fp1) -cost (p, lambda)

Integral Integral dtc=[deaa]; cee=ce; //mtot - total numbers of inputs to the centrallized controller Σ // Split the plant controller sp and find the sizes of each piece
// using the stored data in init.dat. Get the pieces from each submatrix
// which connect input i to output o, that is Moi; Bpa, Bpe, Cap, Cyep
// Some of the indices into the rows and columns used repeatedly are
// precalculated to save processing, ie, mapi-ma+1 etc //np,nka,nke - orders of plant,airframe,engine controllers computes the cost and gradient of the cost function at the current point (parameter vector). The partitioning and tracking costs and lambda - the weighting applied to the tracking cost in the total load 'const.dat' la le ka ma ke me pea nka nke sp sk np nk frq omega //la, le - numbers of airframe and engine y feedbacks //Note some of the sizes of matrices, numbers of inputs and outputs // From the airframe controller ska, get Baa,Baya,Caa,Ceaa, // Daa,Daya,Deaa,Deaya in similar manner to the way that the plant // matrix was split up //the dp submatrix of the plant is //the ap: xp -> xp and does not controller //Generate ska and ske controllers from the parameter vector p cost at point p (current parameter vector) // to be further broken up E 9 2 deaz <sg ng swi nwi swo nwo weight stabil aore Nperf Ntrack; -> ya -> ye dx ^ **2** 92 //ka, ke - airframe and engine outputs //split the airframe î Df0 - gradient of the partitioning cost function î î î da: ya fp0 - value of the partitioning cost function Df1 - gradient of the tracking cost function //ma,me - airframe, engine inputs 263 -> zea ď. cp: xp ĕ ne cb: xb ά cb: xb 88 Ŷ //part of bp: Ŷ î î value of the tracking cost function //part of bp: Df - gradient of the total cost function //kept zero ba: ya //split sp ba: ea × da: ya their gradients are returned seperately. deaya-da([kapl:katot],[mapl:(ma+la)]); // //part of //part of cyap-cp({(mpea+1):(mpea+1a)),:);//part of cyep=cp(((mpea+la+1):(mp)],:); //part of //part of //part of ë const.dat - datafile of constants The function COST in file COST.MTX daa-da((1:ka), (1:ma)); daya-da((1:ka), [map1: (ma+la)]); deaa-da((kap1:katot), (1:ma)); (aa, ba, ca, daj "split (ska, nka); - the parameter vector cep-cp([(ma+pea+1):mpea],:); baya-ba(:,[(ma+1):(ma+la)]); (ap, bp, cp, Df] "split (sp, np); ceap\*cp([map1:ma+pea],:); ceaa-ca((kapl:katot),:); [katot ntmp]-size(ca); map1 "ma+1; mame"ma+me; cost function kapl=ka+l; kp=ka+ke; bpe-bp(:, (kap1:kp)); fp - the total (mp ntmp) -size(cp); caa-ca([1:ka],:); bpa-bp(:, [1:ka]); baa-ba(:,[1:ma]); cap-cp([1:ma],:); [ska, ske] -mat (p); mpea-ma+me+pea; nt =np+nka+nke; mtot =mt +lt; Output: Input: kt-ka+ke; mt =ma +me: lt-la+le; 2222222222222

(inv(sI-A)\*B; I]\*Wi\*((Wo(K- tilda(K))Wi)^T)\*Wo\*(C\*inv(sI-A), I) ) Compute the costs; fp0 (performance), fp1 (tracking) and their gradients fp0-Integral (tr (Wo(K-tilde(K))Wi)) fp1-Integral (tr ((nrm(T-tilde(T)))^T) (nrm(T-tilde(T)))) // Form Teent from Input: Zac to output: Zeac for // command tracking using Centralized Controller. This was created in // start.mtx because it only needs to be defined once (does not vary // Form the Equivalent System tilde(K) using Partitioned Subcontrollers // From the engine subcontroller, get Beea,Bee,Beye,Cee,Desam,Deem,Deye bpa'daa+bpe'deea'deaa, bpe'dee, bpe'deea'deaya+bpa'daya, bpe'deye); // Generate the state-space representation tilde(T) from input: Zac // to output: Zeac, command tracking using Partitioned Controller. ct0=[caa,0\*ones (ka, nke),0\*ones (ka, np);deea\*ceaa,ce, (-deea\*ceap)]; //split the engine controller // Get the Centralized controller K for the performance cost Decompose the input and output weighting functions into (bpa\*caa+bpe\*deea\*ceaa), bpe\*cee, (ap-bpa\*daa\*cap... +bpa\*daya\*cyap+bpe\*deea\*deaya\*cyap+bpe\*deye\*cyep) |; bpa\*caa+bpe\*deea\*ceaa,bpe\*ce, (ap-bpe\*deea\*ceap)]; ctc={ceaa,0\*ones(pea,nke), (-deaa\*cap+deaya\*cyap)}; beea \*ceaa, ae, (-1 \*bee \*cep-beea \*ceap-beea \*deaa \*cap be: zea -> xe de: zea -> ue be: ee -> xe be: ye -> xe ce: xe -> ne de: ee -> ue atc=(aa,0\*ones(nka,nke),(-baa\*cap+baya\*cyap);... -bpe deea deaa cap-bpa deea caap-bpe dee cep bt0=[baa,0\*ones(nka,me),baya,0\*ones(nka,le);.. de: ye -> dt0=[daa,0\*ones (ka,me),daya,0\*ones (ka,le);... btc=[baa;beea\*deaa; (bpa\*daa+bpe\*deea\*deaa)]; // input data originally read from init.dat // from inputs: Ee, Ea, Y to outputs: Ua, Ue. at0\*(aa,0\*ones(nka,nke),0\*ones(nka,np);... //part of // their state space representations Dfp0-partitioning cost gradient =-2\*Re[ (beftFactor) beea deaa, bee, beea deaya, beye;.. [aw1, bw1, cw1, dw1] -split (sw1, nw1); (awo, bwo, cwo, dwo] -split (swo, nwo); deea "deaa, dee, deea "deaya, deye]; +beea\*deaya\*cyap+beye\*cyep);... Dfpl=tracking cost gradient beye-be (:, [ (me+pea+1) :metot]); deye=de (:, [ (me+pea+1) :metot]); (ae, be, ce, de] "split (ske, nke); bee=be(:, { (pea+1): (me+pea) }); dee=de(:, [ (pea+1) : (me+pea) ]); //PARTITIONING COST MATRICES beea\*ceaa, ae, -beea\*ceap;... (ak, bk, ck, dk] - split (sk, nk); (ag, bg, cg, dg) -split (sg, ng); --2\*Re[ (LeftFactor) (Right Factor) )' //TRACKING COST MATRICES (ntmp metot | -size (be); // with the parameter) beea = be (:, [1:pea]); deea-de(:,[1:pea]);

>

```
fPerf which measures the performance cost of approximation and fTrack which
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                // COST evaluates the total cost of approximating a centralized controller
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           hierarchically partitioned subcontrollers. It is the sum of two costs,
costimtx
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   and DfO, Df1 and Df
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                (taj+Qaj-aj
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                DE-DEO+DE1
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      'Df0* (z (nka, nke+ke); eye (nke), z (nke, ke); z (np, nke), bpe; z (ka, nke+ke); z (ke, nke),...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               D[pl=Integral ( [lnv(s1-A)*B; ]]*(nrm*(T- tilde(T))*T*nrm*[C*inv(sI-A), I] )
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  - (deaa·cap+ceap-deaya·cyap), ddaa;z(me,nka+nke),-cep,z(me,ma);z(le,nka+nke),...
cycp,z(le,ma)]'Df1*(z(nka,nke+ke);eye(nke),z(nke,ke);z(np,nke),bpg;..
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           //If the airframe controller is to be optimized, calculate the cost gradients
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   Dfal=((eye(nka),z(nka,nke+np+ma);z(ma,nka+nke),-cap,eye(ma);z(la,nka+nke),...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               // Integrate over the frequency interval [FRQ(1), FRQ(2)] using Simpsons rule,
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      cyap, z (la, ma) | *Df1 * [eye (nka), z (nka, ka+pea); z (nke, nka+ka), beea; z (np, nka),...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             ( [inv(sI-A)*B; I]*WI*((Wo(K- tilde(K))Wi)^T)*Wo*(C*inv(sI-A), I) )
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       z (la, nt+mt), eye (la), z (la, lo) | ·Df0^ (eye (nka), z (nka, ka+pea); z (nko, nka+ka)...
beea; z (np, nka), bpa,bpe^deea; z (ka, nka), eye (ka), z (ka,pea); ...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      Dfe0= {{z (nke, nka), eye (nke), z (nke, np+mtot); ceaa, z (pea, nke), -ceap, deaa,...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                z (pea, me), deaya, z (poa, le); z (me, nt+ma), eye (me), z (me, lt); z (le, nt+mt+la),...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     //with respect to the parameters belonging to the airframe controller by //multiplying DFO and Dfl by the corresponding left and Right Factors
                                                                                                                                                   ( [inv(sI-A)*B; I]*(nrm*(T- tilde(T))*T*nrm*(C*inv(sI-A), I) )
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             ktrc=nrm*(cg*inv(om*)ay*eye(ng)-ag)*bq+dq-ctc*k1*btc-dtc);...
fpl=fpl+welght(1)*om*sum(dlag(ktrc'*ktrc));...
Dfl=Dfl+welght(1)*om*[k1*btc;eye(ma)]*ktrc'*nrm*[ctc*k1, eye(pea)];...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           //engine parameters is DfeO and gradient of the tracking cost is Dfel.
//DfeO=[ (LeftEactor) DfO (RightFactor) ]'
//Dfel=[ (LeftFactor) Dfl (RightFactor) ]'
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        //Right Factors. Gradient of partitioning cost with respect to the
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           //This is done for both the partitioning cost DfaO and the tracking //cost DfeO (with respect to the airframe parameters) //DfaO-{ (LeftFactor) DfO (RightFactor) ]' //Dfal-{ (LeftFactor) DfI (RightFactor) ]'
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     1f aore<>>1,...
Diao=(leye(mka),z(mka,nke+np+mtot);z(ma,nt),eye(ma),z(ma,me+lt);...
Diao=(leye(mka),z(mka,nke+np+mtot);z(mke,nk
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      Df0-Df0+weight (1) *om/Nperf (1) * [k0*bt0; eye (mtot) ] *wi*kdiff' *wo*...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       //calculate the cost gradients with respect to the engine
//parameters by multiplying DfO and Df1 by the requisite Left and
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  wo-cwo*inv (om* jay*eye (nwo) -awo) *bwo+dwo; ...
kdiff=wo* (ck*inv (om* jay*eye (nk) -ak) *bk+dk-ct0*k0*bt0-dt0) *wi;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        Build up the partitioning costs fp0, fp1, and the integral
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               Dfal=({z (nke, nka), eye (nke), z (nke, np+ma); ceaa, z (pea, nke),...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            fp0=fp0 +weight(1) *om*sum(diag(kdiff'*kdiff)/Nperf(1));...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                //Similarly, if the engine controller is to be optimized,
                                                                                                                                                                                                                                                                                                                                                                                                                           Df0=0*ones(nka+nke+np+ma+me+la+le,nka+nke+np+ka+ke);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     coefficients are in the variable WEIGHTS..
                                                                                                                                                                                                                                                                                               // Initialize the costs and gradients to zero
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               w1=cw1*inv(om*jay*eye(nw1)-aw1)*bw1+dw1;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    bpa, bpe*deea; z (pea, nka+ka), eye (pga) [)';...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                         Df1=0*ones (nka+nke+np+ma, nka+nke+np+pea);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       nrm-dlag((lambda./Ntrack(1,:)) **.5);...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              else Dfa0=0*ones(ska); Dfal=Dfa0; end
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       part of the gradients DfO and Dfl
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 for 1=1:frq(3,1), om-omega(1);...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      k1=inv (om* jay*eye (nt) -atc); ...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        k0=1nv (om* jay*eye (nt) -at0); ...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        z (ke, nka+ka), deea])';...
                                                                                                                                                                                                             (RightFactor) )'
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        [ct0*k0, eye (kt)];...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         Df0-Integral
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  if aore<>2,
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               ::::
```

else Dfe0-0.ones(ske);Dfe1-Dfe0;end z (pea, nke+ke) ])';...

//described earlier and the airframe and engine costs and gradients //Assemble the final cost and gradients using the formulas coef=(log(frq(2,1)/frq(1,1)))/(3\*frq(3,1)); fp0=real(coef\*fp0); fp1=real(coef\*fp1); Df0=-2\*coef\*real (longcol (Dfa0, Dfe0)); Df1=-2\*coef\*real (longcol (Dfa1, Dfe1)); //just generated.

// is the parameter vector, p, and the tracking weighting vector, lambda measures the error in meeting the command tracking requirement.

// The file 'const.dat' contains the constants of the process
// the dimensions --- ka,ma,ke,me,pea,np,nka,nke,nk,nwi(o),ng,la,le(k-outputs,
// the dimensions --- ka,ma,ke,me,pea,np,nka,nke,nk,nwi(o),ng,la,le(k-outputs,
// the range vector of frequencies --- frq
// the actual frequencies --- omega
// the plant in system form --- sk
// the plant in system form --- sk
// the global controller in system form --- sg
// the global tracking command matrix in system form --- sg
// the global tracking command matrix in system form --- sg
// the mormalization matrix for the performance cost --- Nperf
// the normalization matrix for the tracking cost -- Ntrack
// the normalization matrix for the tracking cost -- Ntrack
// the state space representation for the weighted difference between global
// controller and assembled partitioned controller is determined. The
// morm of the resulting system is computed.The norm is calculated by
// applying a Simpson's rule suitably modified to account for the exponential
// distribution of the omegas.

transfer matrix and the nominal one is also calculated. This is added to The outputs are fp0 --- the H\_2 norm of the weighted difference between fpl --- the normalized H 2 norm of the deviation from command centralized and partitioned controllers The normalized H\_2 norm of the difference between the tracking the previous H\_2 norm. The gradient of this sum is computed. nominal command tracking - fp0 + fpl đ

at Created by Phil Schmidt and students Nader Kamrani and Brian Holawecky The University of Akron with the support of NASA Lewis Research Center // Created by Phil Schmidt
// The University of Akron
// under grant NAG-3-1146.

--- the gradients of fpl, fpl and fp resp.

# linesrch.mtx

```
aold-a; faold-fa; fpaold-fpa; a-alnew; fa-fal; fpa-fpal;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            else if abs(fpal)<=-Sigma*fpz,flag=0;ali=alneW;retf; end;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               If 0<= (b-aold) *fpal, b-aold; fb-faold; fpb-fpaold; end;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             elseif fal>fa,b=alnew;fb=fal;fpb=fpal;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 gradlent.
                                                                                                                                                                                                                                                                                                                                                                                                                                                         al-alnew;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            albound-min((fmin-fz)/(Rho*fpz),alphamax); // search restricted to [0,albound]
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    1 - solution with function value less than fmin
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       // almost directly from Fletcher, PRACTICAL METHODS OF OPTIMIZATION, Wiley, // 1987 pp.34,35. The only part of the code not in the book is // the minimization of the cubic interpolation of fal, fpal, // and fpalp used to generate the next alpha in both the bracketing // and the sectioning phases
                                                                                                                                                                                                                                                                                                                                                                                           dfdpz - [real, vec] directional derivative of objective function at al=0
d - [real, vec] the direction vector which the BFGS procedure
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   if fal>[al>falpr, a=alpr;b=al;fa=falpr;fb=fpalpr;fpb=fpal;exit;end;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        follows
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             ... cas--min cnen tiag-i; ail-ai; reti; end;...
If fal>fa+al*Rho*fpz,a-alpr;b-al;fa-falpr;fb-fal;fpa-fpalpr;fbb-fpal;...
                                                                                                                                                                                                                   of the unconstrained optimization process. Given the search direction,
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   (x+al*d); this is included to help keep the number of computations
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   - reached end of search interval, reduction without finding min
                                                                                                                                                                                                                                                a point is found that satisfies the Armijo-Goldstein conditions using
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     to the univariate problem. fal, fp0, fp1 – {real,scal} the cost function values at the point
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 flag - (int, scal) denotes the condition with which INACCURATE has
completed the line search.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  alphamax - [real,scal] maximum alpha set be stability constraint.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           if a point is found such that the objective value is less than
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    // slope when alpha=0, ie directional slope
// alpha(1-1) the previous alpha
// function value at previous point
                                                                                  (all, fal, fp0, fp1, dfdpal, dfp0, dfp1, flag| =INACCURATE(x, fz, dfdpz,...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     // The following algorithm is not documented in great detail as it
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              all - [real, scal] the step-size determined to be the solution
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      bounds=[1;-10*deltaf/fpz;alphamax]; // used to find inital alpha
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 fmin - (real, scal) user-defined lower bound on the problem;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           // The Armijo/Goldstein parameters, described in more detail in
// Fletcher, PRACTICAL METHODS OF OPTIMIZATION, Wiley, 1987.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    dfdpal,dfp0,dfpl - (real,vec); gradlents at the min point.
                                                                                                                                                                                                                                                                                                                                          \kappa=\{real,vec\} the current point corresponding to al=0 fz - \{real,scal\} objective function value for al=0.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     [dfdpal, dfp0, dfp1, fal, fp0, fp1] "COST (x+al*d, lambda);
                                                                                                                                                                                                    performs Fletchers inaccurate line search as part
                                                                                                                                                                                                                                                                                                                                                                                                                                                         has chosen to perform the univariate search.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               // slope at previous point
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         deltaf - [real, scal] estimated change in cost
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          fmin, then the procedure will terminate.
                                                                                                                                                                     The function INNACURATE in file LINESRCH.MTX
                                                                                                                                                                                                                                                                                       a two stage bracketing/sectioning procedure.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           al, fal, fpal, alpr, falpr, fpalpr,...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          // initial alpha
d, fmin, deltaf, lambda, alphamax)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 - solution found;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 // The bracketing phase
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       fpal-dfdpal' d;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  to a minimum.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              while 1>0 do ...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             : (spunoq) uju=[e
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        exit; end; ...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 ;p* ,zdpJp*zd;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       fpalpr=fpz;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         Output:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      S1qma=0.7;
                                                                                                                                                                                                                                                          Rho-0.01;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            rau2=0.1;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          falpr-fz;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        Tau3-0.5;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                raul-10;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              alpr=0;
```

out Similarly, in the sectioning phase, construct a cubic interpolant using the values of f(A), f(B), f'(A) and f'(B) and find a minimum point which (hopefully) lies in the interval [(i-Tau2)\*A+ Tau2\*B,Tau3\*A+(i-Tau3)\*B), O<Tau2<Tau3<1. The new f(alpr) and f'(alpr). Note that the new point is chosen by moving far to the right in an effort to bracket an interval of acceptable points. if abs(fpal)<---Sigma\*fpz, flag=0;all=al;retf;end;...
if fpal>=0, b=alpr;a=al;fb=falpr;fa=fal;fpb=fpalpr;fpa=fpal;exit;end;...
if abs(al-albound)<0.001\*albound, flag=2;all=al;retf;end;...</pre> a cubic polynomial which interpolates the values of f(al), f'(al), else alnew=alpr-fpalpr/(2\*c3);end;...
elself c3\*c3-3\*c4\*fpalpr<=0, alnew=-{abs(fpalpr)/fpalpr)\*albound;...
else alnew=al+(sqrt (c3\*c3-3\*c4\*fpalpr)-c3)/(3\*c4); end;... else if alnew<-n, alnew-n; elseif alnew-m, alnew-m; end, end;... point is chosen in this interval (which is a subinterval of its [al+(al-alpr),al+Taul\*(al-alpr)]. alpha is chosen to minimize if m<n, if alnew<=m, alnew=m; elseif alnew>=n, alnew=n; end,... else m=2\*al-alpr; n=min({albound,altTaul\*tal=alpr)});end;...dal=al-alpr; dfal=(fal-falpr)/dal; dfpal=(fpal-fpalpr)/dal;...c4=(dfpal-2\*(dfal-fpalpr)/dal; c3=0.5\*dfpal-1.5\*dal\*c4;...if c4==0, if c3==0, alnew==(abs(fpalpr)/fpalpr)\*albound;... predecessor). Eventually an acceptable point will be found. c4=(dfpal-2\*(dfal-fpa)/dal)/dal; c3=0.5\*dfpal-1.5\*dal\*c4;... c4\*\*0, if c3\*\*0, alnew\*b; else alnew\*a-fpa/(2\*c3);end;... [dfdpal, dfp0, dfp1, fal, fp0, fp1] =COST(x+alnew\*d, lambda);... if albound<=2\*al-alpr then m=(al+albound)/2; n=albound;... elseif c3\*c3-3\*c4\*fpa<-0, alnew-b;... else alnaw=a+(sqrt (c3\*c3-3\*c4\*fpa)-c3)/(3\*c4);end;... The steps above select a new alpha in the interval if fal>fz+Rho\*alnew\*fpz, b=alnew;fb=fal;fpb=fpal;... if alnew<"m, alnew=m;elseif alnew>=n, alnew=n;end,... while 1>0 do ... a, fa, fpa, b, fb, fpb, ... dal-b-a; dfal-(fb-fa)/dal; dfpal-(fpb-fpa)/dal;... if fal<-fmin , flag=1; all=alnew; retf; end; .. alpr-al; falpr-fal; fpalpr-fpal;... m-a+tau2\*dal; n-b-tau3\*dal;... The sectioning phase fpal-dfdpal' d;...

It makes calls to the user-defined function (UDF) The function INACCURATE in the file 'linesrch.mtx' performs Fletcher's inaccurate line search as a part of the unconstrained optimization COST in 'cost0.mtx' which produces both the function value and procedure, PARTITION.

at each line search in the overall optimization process, that INACCURATE will solve the univariate minimization problem

This is accomplished by use of a two-stage (bracketing/sectioning)

only to find a point which satisfies the Armijo-Goldstein conditions. determining a search direction has proven to be robust enough to procedure which employs cubic approximation that will seek This approach is useful because the BFGS method of be effective even with an inaccurate line search.

10:22:08 93/07/02

thorough description of how each parameter is used in this procedure, aspects of the search computations. These parameters have been set to values that have proven to be adequate for this procedure. Of course, these may be changed to suit the user's needs. For a more This line search makes use of many parameters that define various see Fletcher, PRACTICAL METHODS OF OPTIMIZATION, Wiley, 1987 \*\*\*\*\*\*\*\*\*\*\*

Parameters: The Armijo/Goldstein parameters should satisfy:

0 < Rho < 1/2 --- smaller is easier to satisfy; we use 0.01 Rho < Sigma < 1 --- Larger is easier to satisfy; we use .7 The next is used in the bracketing stage to find (if necessary) larger and larger intervals which might contain an acceptable, i.e. Armijo/Goldstein, interval:

Taul > 1 --- we use 10, the larger, the more effort might be used in the sectioning phase.

The last two parameters are used to reduce the acceptable interval until an acceptable point is found:

0< Tau2 < Sigma is advised --- we use 0.1 Tau2 < Tau3 <= 1/2 --- we use .5 to get the greatest reduction of size.

Flag is used by INACCURATE.DAT to characterize the condition of the line search when it terminates. The values can signify: nothing special to note about the solution. flag=1 - a point has been found that gives an objective value  ${\bf r}$ Notes on the use of the output variable 'flag' in this function: flag-0 - the solution to the line search has been found;

conditions. It should be noted however that the new point will have a reduced function value (although, perhaps endpoint of the interval without satisfying the A-G flag-2 - the bracketing step has converged to the right hand insufficient to trigger the A-G conditions). less than fmin.

Programmer: Steven Ims Created: 01/30/90

Rewiltten by Phil Schmidt and students Nader Kamrani and Brian Holawecky at The University of Akron with the support of NASA Lewis Research Center :::::

under grant NAG-3-1146.

{sm} =mod1 (s, ns)

```
form where A has 2x2 companion matrix blocks whose first rows are [0 1] and
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        // If two real elgenvalues are still very close in value, shift one slightly
                                                               The inputs are a system matrix and its order. The system is put into the
                                                                                                                                                                                       remain fixed (in case of zero entries in Bi, the entry is fixed at le-9).
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        // eigenvalues at the top of am and sends all the 2x2 matrices representing \lambda // the complex eigenvalues to the bottom of am.
                                                                                                                          whose second rows are (a b). The transformation matrix is normalized by
The function MODL in file MODL.MTX puts a system matrix into a modified modal form reducing the number of parameters in the optimization.
                                                                                                                                                            requiring that all nonzero entries in the first column of the B matrix
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      Convert the input system matrix S into the internal Matrixx modal form
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 // are on the diagonal and the complex conjugate eigenvalues (a + bl) // and (a - bl) are stored in 2x2 matrices of the form (a bl on the first // row and [-b a] on the second row.
                                                                                                                                                                                                                          The C, and D are full. This should work except in the special case of
                                                                                                                                                                                                                                                                                                                                                                                                                         sm - the system matrix, now in modified modal form, still has order
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            q-mod(nx,2); nrhalf-(nr-q)/2; tpnt-j+2; mpnt-j+nrhalf+1;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         // Since S=\{A,B,C,D\} is the form of a system matrix, multiplying // \{P,\ I\} * $ * [inv(P); I] will have the following effect on A,B,C,D
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         // where the A submatrix is of the form that the real eigenvalues
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    // Rearrange the real elgenvalues so that the ones that are close
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                else,...
m=p([i:i+1],:);...
if l>j+1, for k=i:-1:j+2, p(k+1,:)=p(k-1,:); end; end;...
p([j+i:j+2),:)=m;...
                                                                                                                                                                                                                                                                                                                               s - the system matrix to be put in modal form
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       or 1=1:nr=1, ...
if a(i,1)=a(i+1,1+1), a(i,1)=a(i,1)-10e-9; end;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    for 1=1:(nrhalf=1), temp=p(mpnt,:);...
p({(tpnt+1):mpnt],:)=p([tpnt:(mpnt=1)],:);...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               p(tpnt,:)-temp; tpnt-tpnt+2; mpnt-mpnt+1;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      Split the modal form sm into its components
                                                                                                                                                                                                                                                                                                                                                               ns - the order of the input matrix
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      (am, bm, cm, dm] -split (sm, ns);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       // In value are separated.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          1f am(1,1+1)=0,1=1+1;...
                                                                                                                                                                                                                                                                     repeated real roots.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      1-1; J-0; p-eye(ns);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            [sm,t]=modal(s,ns);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          j=j+2; 1=1+1;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              // more negative
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   while 1 < ns, ...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                (n, o) =size (cm);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    a-p-amtinv(p);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    for 1-1:nr-1,
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 c-cm* inv (p);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          If nr > 2,
                                                                                                                                                                                                                                                                                                                                                                                                  Output:
                                                                                                                                                                                                                                                                                                        Input:
                                                                                                        22222222
```

## modI.mtx

```
// returning the desired structure [0 1; a b] which have the same
                                   ö
   // Construct a transformation matrix T such that
// T * a * inv(T) has the effect of taking 2x2 blocks of
// real eigenvalues (or of complex conjugate pairs) and
Construct a transformation matrix T such that
                                                                                                                                                                                                                                                                                     t (2*1,2*1) =1; t (2*1-1,2*1)=1; ...
t (2*1,2*1-1) =a (2*1-1,2*1-1) =a (2*1-1,2*1); ...
t (2*1,2*1) =a (2*1,2*1) =a (2*1,2*1-1); ...
                                                                                                                                               // elgenvalues as the 2x2 blocks.
                                                                                                                                                                                                                                                           for 1-1: (ns-q)/2,
                                                                                                                                                                                                                       q~mod (ns, 2);
                                                                                                                                                                                      t =0 *eye (ns);
```

// The variable q keeps track of an odd size matrix, i.e. an odd entry which // does not fit in the 2x2 blocks if q=1, t (ns, ns)=1; end;

// [T I] \* system matrix \* [inv(T); I] will generate the following aa=t\*a\*tinv; bb\*t\*b; cc=c\*tinv; tinv=inv(t);

// Set the first entry in the modal blocks to zero; {0,1;a.b for 1=1:(ns-q)/2, aa(2\*1-1,2\*1-1)=0; end; // Construct the system matrix sm which is now in the modified modal form sm=[aa, bb;cc, dm];

// Created by Phil Schmidt and students Nader Kamrani and Brian Holawecky // The University of Akron with the support of NASA Lewis Research Center // under grant NAG-3-1146.

[ska, ske] =mat (b)

# pamat.mtx

```
reconstruct the partitioned system matrices ska and ske. It allows for the possibility that one of the subcontrollers is fixed and loads its initial value. Note that the constant submatrices Bal, Bel, Daa, Daya, Deac, Dee and
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        Created by Phil Schmidt and students Nader Kamrani and Brian Holawecky at
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              // This program takes the parameter vector P_{\nu} which is a column vector, and // uses the dimensions of the subcontrollers stored in 'const.dat' to
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          The University of Akron with the support of NASA Lewis Research Center
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  Deye are properly loaded.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                under grant NAG-3-1146.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              else ske-ske0; end,
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     else ska-ska0;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       // be used.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            // Set the counter, first to 0 then to mka etc. as new blocks are built from p
                                                                                              the parameter vector p. Note that p is: Aa,Ae,Ba,Ca,Deaa,Deaya,Be,Ce,Deea. The other submatrices are all constants and are loaded from const.dat
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          // and copy the requisite entries from the parameter vector p. Recall the // and copy the requisite entries from the matrices (refer to modil modal from of the system matrices (refer to modil.mtx) reduces the // A submatrices to 2x2 companion blocks of the form {0,1; a,b}, so only // the a and b need to be read if aore<>1, aa=0*eye(nka); q=mod(nka,2); ...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 // If the engine controller is not fixed, load the first column of be // from the bel entry stored in const.dat. Then load rest of be,ce,deea // from p. Construct de from deea, dee (and deye depending on whether
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           // If airframe is not fixed, then create a blank aa (A submatrix for ska)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        // Note that the first column of the ba entry is fixed as required by // the modal form, so load it from bal (saved in const.dat)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         // entries from the p vector, with the same provisos as for aa above
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 // If engine is not fixed, then create a blank ae and copy requisite
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          // If airframe is not fixed, generate the ca,deac,da entries from p.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   for 1=1:matot, deac(:,1)*p{{dum+1:dum+pea],1); dum-dum+pea; end;..
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   for 1=1:nka, ca(:,1)=p({dum+1:dum+katot|,1}; dum-dum+katot; end;..
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              for i=2:matot, ba(:,1)=p([dum+1:dum+nka],1); dum*dum+nka; end;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   for i=1:matot, daaya(:,1)=p([dum+1:dum+ka],1); dum=dum+ka;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   // The rest of ba, ca, deac are loaded from p. If Ds are not // their values are read from p, otherwise from constants // (and daya depending on whether feedback lya exits). If aore<>1, ba(;,1)-ba1;..
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   else if lya=0, daaya=daa; else daaya=[daa, daya]; end;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       me ke lye bel
                                                            the partitioned system matrices ska and ske from
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  ав(2*1-1,2*1)=1; ав(2*1,2*1-1)-р(dum+2*1-1,1);
                           The function MAT in file PARMAT.MTX generates
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            aa(2*1-1,2*1)-1; aa(2*1,2*1-1)-p(2*1-1,1);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              1f aore<>2, ae=0*eye(nke); q=mod(nke,2); ...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       load 'const.dat' nka ma lya nke bal ka pea
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                1f q=1, ae (nke, nke) =p (dum+nke, 1); end;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        matot-ma+lya; katot-ka+pea; dum=nka;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     daa daya dee deye ska0 ske0 aore flxd
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  // load the default data from const.dat
                                                                                                                                                                                                                                                         const.dat - datafile of constants
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          if q=1, aa (nka, nka) =p (nka, 1); end;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              ae (2*1, 2*1) -p (dum+2*1, 1); end; ...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        metot=me+pea+lye; dum=dum+nke;...
                                                                                                                                                                                                                                                                                                                             ska - the airframe controller
                                                                                                                                                                                                                        parameters
                                                                                                                                                                                                                                                                                                                                                                         ske - the engine controller
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              aa(2*1,2*1)-p(2*1,1); end;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               1f (flxd-1)*(flxd-2) <> 0, ...
                                                                                                                                                                                                                        p - the vector of
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          for 1=1: (nka-q)/2,...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          for 1-1: (nke-q)/2,...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      da=(daaya; deac);...
                                                                                                                                                                                                                                                                                                  Output:
                                                                                                                                                                                      Input:
********
```

```
// If the airframe controller is not fixed, then construct ska from aa,ba
// ca and da. If there is no feedback defined lya=0, then extend
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               // ska by one column of zeros to accomodate the dummy feedback that will
                                                                                                                                                                                                                       else for i=1:pea, deea(:,i)=p((dum+1:dum+ke],l); dum-dum+ke; end;...
                                                                                                                                                                                                                                                                                                                                                                   dum-dum+ke; end;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       // If the engine controller is not fixed, then construct ske.
// If there is no feedback defined lye=0, then add the zero column
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       // If the engine controller is fixed, then let ske=skeO the fixed
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          if aore<>2, ske=[ae,be;ce,de|;...
if lye=0, [rske,cske| =size(ske); ske=[ske,0*ones(rske,1)]; end,
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   if aore<>1, ska=(aa,ba;ca,da);...
if lya=0, {rska,cska|=size(ska); ska=(ska,0*ones(rska,1));end,
                                                                              for 1=2:metot, be(:,1) =p([dum+1:dum+nke],1); dum=dum+nke; end
for 1=1:nke, ce(:,1) =p([dum+1:dum+ke],1); dum=dum+ke; end;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          // If the airframe controller is fixed, then use ska-ska0
                                                                                                                                                                                                                                                                                                                          if (fixd-1)*(fixd-2) <> 0,meplye=me+lye;...
for i=1:meplye, deeye(:,i)*p({dum+1:dum+ke},1); dum=else if lye=0, deeye=dee; else deeye=(dee, deye); end;
                                                                                                                                                                                 1f (fixd-2)*(fixd-3)=0, deea=0*ones(ke,pea);...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      // to accomodate the dummy feedback.
// there is any feedback lye).
                                             if aore<>2, be(:,1) -bel;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             // engine controller.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    de=(deea, deeye);...
```

### 10:23:53

Inputs: START \$ di--dfx\*hi\*dfx\*gradl;...// search direction is -Approx Inv Hessian\*gradient (all, fp\_11, j0, j1, grad11, grad0, grad1, flag| "INACCURATE (p\_1, fh (cnt), grad1,... checked by function CONVERGE) are met or some other stopping criteria It implements the Broyden-Fletcher-Goldfarb-Shanno method of finding // prestore the messages to save the length of loops, in order that the a search direction for minimization, then calls function INACCURATE to generate a new point by minimizing in that direction. This are met (number of iterations exceeds maximum or function value is p.1 - the ith point along the optimization process inter.dat - the datafile where intermediate results are stored a ph='A phat has been found that gives an objective value < fmin.'; process is then repeated until the convergence conditions (as const.dat - the datafile of constants inter.dat - the datafile containing intermediate results load 'const.dat' omega nka nke atot stabil load 'inter.dat' p\_l gradi gradi gradi fh jhû jhi hi lambda [row,col]=size(p\_l); [cnt,tmp]=size(fh); // While the convergence/stopping conditions have not been Y-[Ch (ITER), Jh0 (ITER), Jh1 (ITER) ], PLOT (ITER, Y, plpar);.... The function PARTITION in file PARTITIO.MTX is the alphamax=le16;...// check on stability constraints // Read stopping conditions from the vector stop dfx-diag(ones(row,l)-{fx;0\*ones({row-atot),l}); delf-max(0.9\*abs(fh(cnt)),l0\*\*(-6)); outer level loop in the optimization process. stop - the vector of stopping conditions 1s-'LINESEARCH TERMINATED BY HITTING BOUND'; maximum number of iterations'; 1f (stabil-p i(I))/di(I)<alphamax,...</pre> alphamax=(stabil-p\_1(I))/d1(I);... A pipar - STRIP YLAB/f(P)|f0(P)|f1(P)/'; lf cnt>20, ITER=[cnt-20:cnt]'; // Matrixx line limit be avoided not = NOTICE: STABILITY BOUND.'; eps1 -stop(1) ones(p\_1); else d1(I)=0; end,... else ITER = [1:cnt]';. while cvflg<0.5 do... p\_1-partition(stop) // The function PARTI
// outer level loop 1
// It implements the
// a search direction
// to generate a new
// process is then re
// checked by functio
// are met (number of
// laput:
// Input:
// Input:
// output:
// inter.dat - the
// output:
// p\_l - the ith po
// inter.dat - the
// inter.dat - the cnt, fh (cnt), ... for I-l:atot,... 1f d1(1)>0,... 1f fx(I)=0,... maxent -stop(4); eta = stop(3); fmin =stop(5); 1f cnt>1, .. del -stop(2); ex=, exceeded cvgfact-1; cvf1g-0;

## partitio.mtx

Parameters being defined in 'param' vector: any defaults were assigned in PARTITIO.MTX is an implementation of the Broyden-Fletcher-Goldfarb-Shanno (default=le-9) - column vector with element-wise convergence |grad f(p\_11)| < eta; - maximum number of iterations of procedure line search being used to locate the the next point in UDF INACCURATE. It makes calls to the user-defined functions INACCURATE and CONVERGE in the files 'linesrch.mtx' and 'converge.mtx' resp. hi-pomat hi\* (pomat') + ((psi\*psi')/psiQi);...// updates Inv Hessian Approx method of determining a search direction, with Fletcher's inaccurate It also uses the MATRIXX PLOT function to show intermediate results. fh jh0 jh1 gradi grad0 grad1 lambda fx h1;end... elseif gradil(I)>0, fx(I)=0; for J=1:row,hi(I,J)=0;hi(J,I)=0;end,... save 'inter.dat' p\_i fh jh0 jh1 grad1 grad0 grad1 lambda fx h1;... i isag=i,uispia\_pui; ... save 'inter.dat' p\_l fh jh0 jh1 grad1 grad0 grad1 lambda fx h1;... stop = (real, vec) column vector of stopping conditions save 'inter.dat' ont p\_i fh jh0 jh1 gradi grad0 grad1 lambda fx h1; PARTITION is the top-level function in the optimization routine cvflg=CONVERGE (p\_11, fh (cnt+1), p\_1, fh (cnt), gradil, cvgfact \*epsl, ... limits for elements of ip il - convergence of objective values PARTITION.MTX will solve hierarchical partitioning problems disp('Convergence of I-value and parameters has occurred');
if cvflg=2, disp('all partials are also less than'); eta, end; (default=le-9) - convergence of gradients, di, fmin, deif, lambda, alphamax);...// calls the linesearch If (p 11) - f(p 1) |; if fx(I)=0, if p il(I)>=1.001\*stabil, fx(I)=1;... if all=alphamax, cvgfact=0; display(not); end,... stop = (epsl del eta maxent fmin) cvgfact\*del,eta,fx);...// checks convergence end, end ... // updates stability constraints delf=max(fh(cnt)-fh(cnt+1),10\*\*(-6));... p 11 (I) =stabil; dfx (I, I) =0; end... pQmat=(eye(row)-ps1\*Q1'/ps1Q1);... (default=le-9) if flag-2, display (1s); end;... (default=100) hi (I, I) =1; dfx (I, I) =1;... 1f cnt=maxcnt, disp(ex);... Qi=dfx\* (gradil-gradi);... save 'inter.dat' p\_i if flag=1, disp(a\_ph); p il-p itali'di; ... fh (cnt +1) = fp 11;... 1f mod (cnt, 20) -0, for I=1:atot, ... Jh1 (cnt+1) = j1;... ps101-ps1' \*Q1;... gradi-qradil;... maxcnt psi~ali\*di;... retf; end, ... ret f; end; ... cvafact=1;... cnt-cnt+1;...

fmin

value less than fmin will be considered a solution. Since we minimize a norm, the default is zero. Any point, x, found with an objective allowed in searching for a solution. - lower bound on the objective function. (default=0)

Notes on the use of 'flag' in this function:

characterize the objective function and the current point, x. Flag is used by the INNACURATE functions to

Goldstein conditions (see 'linesrch.mtx'); the precedure It can take on values as follows: (1) flag=0 - a point has been found which satisfles the Armijo-

less than fmin. Procedure terminates.
(3) flag-2 - a point has been returned by INACCURATE which doesn't satisfy the A-G condition, but which corresponds to the maximum allowable step in the search direction. This can (2) flag\*1 - a point has been found that gives an objective value continues.

as though the A-G conditions were satisfied. If problems happen because the steps are constrained to maintain subcontroller stability. Currently, the code continues such as looping are encountered, the user might try to reset HI to the identitly at this point.

The program terminates properly under one of the following conditions:

flag=1 - the objective value is less than fmin.
cnt-mxcnt - the maximum number of iterations has occurred (1) flag-1

- the max deviation in parameters is less than epsi and cvflg-1 23

- same conditions as cvflag-1 and max deviation in the max deviation in COST is less than delta (4) cvflg=2

**57** 

partials is less than eta.

On termination (of any type) the following can be found in the file 'Inter.dat': 

 last parameter vector (can use procedure MAT in 'parmat0.mtx' to generate the corresponding SKA and SKE)

fh - complete total cost history (all iterations) jh0, jh1 - complete cost histories of fPerf and fTrack costs

grad1 - gradient of total cost at last parameter point, grad0, grad1 - gradients of feerf and frack at last point

- last inverse Hessian updated according to values of p\_1, etc. Z

- vector indicating which entries in Aa and Ae are fixed lambda - lambda vector fx - vector indicat

stability bounds

Created: 01/23/90 as BFGS.DAT

Revisions by Phil Schmidt and students Nader Kamrani and Brian Holawecky at The University of Akron with the support of NASA Lewis Research Center Programmer: Steven Ims

under grant NAG-3-1146.

:::::

## 16:00:29

```
// consisting of 2x2 companion matrices (0,1; a,b], only the a,b need to // be included into the parameter vector. The last block of aa may contain // only (a,b) and needs to be included if a is odd (q-1)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             // If airframe controller is not fixed then copy entries that need to be
// optimized from aa. Note that because aa has the modified modal from
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          // If airframe controller is not fixed, then copy ba (note first column
// of ba is fixed and is not copied), ca and da into p
                                                                                 creates the long column vector of parameters p from the partitioned
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                // If engine controller is not fixed, put the required elements of
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             // define sizes of the parameters
                                                                                                                                                                                                                                                                                                                                                                                                                                                   // split ska and ske into their
// component [A,8;C,D]
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         for 1=1: (nka-q)/2, p=[p;aa(2*1,2*1-1]; aa(2*1,2*1)]; end:...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     for 1-1: (nke-q) /2, p-[p; ae (2*1, 2*1-1); ae (2*1, 2*1)]; end;
                                                                                                                                                                                                                                                                                                                                                                                          load 'const.dat' nka nke me pea lye ma lya ka aore flxd
                                                                                                                                                                           is a - the airframe controller state-space matrix ske - the engine controller state-space matrix
                                                                                                                                                                                                                                                                                                      p - the long column vector of the parameters
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         // initialize p
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 // ae into p (same modified modal form as aa)
                                                         The function LONGCOL in file PARVEC.MTX
                                                                                                                                                                                                                                    const.dat - datafile of constants
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           if q=1, p=[p;ae(nke,nke)]; end;...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               if q=1, p={p;aa(nka,nka)]; end;..
                                                                                                                     system matrices ska and ske
                                                                                                                                                                                                                                                                                                                                                                                                                                                         (aa,ba,ca,da)*split (ska,nka);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    (ae, be, ce, de] *split (ske, nke);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        1f aore<>1, q"mod(nka, 2);...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  1f aore<>2, q=mod(nke, 2);...
p-longcol (ska, ske)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  metot=me+pea+lye;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 if aore<>1,...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      matot-ma+lya;
                                                                                                                                                                                                                                                                             Output:
                                                              ::::::::
```

for i=2:matot, p=[p;ba(:,i)]; end;..
for i=1:nka, p=[p;ca(:,i)]; end;..
if (fixd-1)\*(fixd-2) <> 0, for i=1:matot, p=[p;da((1:ka),i)]; end; end;...
for i=1:matot, p=[p;da((ka+1:ka+pea),i)]; end;...

// If the engine controller is not fixed copy be (first col fixed), ce and

for i=1:nke, p=[p;ce(:,1)]; end:...
If (fixd-2)\*(fixd-3) <> 0, for i=1:pea, p=[p;de(:,1)]; end; end;...
If (fixd-1)\*(fixd-2) <> 0, for i=pea+1:metot, p=[p;de(:,1)]; end; end;...

for i=2:metot, p=(p;be(:,i)|; end;...

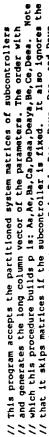
if aorec2,...

// de into p

// Get rid of the 0 which was the 1st entry in p (used to initialize p)

: (d) θ2| S= [λ x]

p\*p((2:x],1);



submatrices which are constants --- Bal, Bal, Daa, Day, Dee, and Deye.

Created by Phil Schmidt and students Nader Kamrani and Brian Holawecky at the University of Akron with the support of NASA Lewis Research Center // Created by Phil Schmiou // The University of Akron // under grant NAG-3-1146,

## restart.mtx

// under grant NAG-3-1146. or a scalar!'), retf;  $m{A}$  // Define UDF PARTITION which is the outermost loop and implements the // Broyden-Fletcher-Goldfarb-Shanno technique for finding a search direction // Define UDF INACCURATE which implements Fletchers inaccurate line search // Verify that the newlamda provided is a row vector or a scalar and is // consistant with pea [riambda,clambda]=size(lambdanew); if riambda > 1, disp('ERROR: lambda must be a row vector or a scalar!'). lambdanew - a new value for lambda, the weighting of the tracking cost as part of the total cost function stop - (optional) the vector of stopping conditions newh - (optional) if any parameter whatsoever is passed, this sets intermediate results at the last checkpoint or the and their sum, as well as the respective gradients for a specified // Define UDF CONVERGE which chacks if convergence conditions are met // Specify the format for displaying matrices // Define the UDF MAT which generates the SKA and SKE subcontrollers // Define UDF LONGCOL which generates the parameter vector from the // SKA and SKE groupings of the parameters restart the optimization process after it has been intentionally halted or has stopped due to the stopping criteria. // Define UDF COST which evaluates the performance cost, tracking // Define UDF 2 which returns a zero matrix of the desired size resize('sstack',1000000); // load the relevant data load 'const.dat' nka nke pea lya lye; atot-nka+nke; load 'inter.dat' p\_i fh jho jhi gradi grado gradi hi fx lambda; final results if convergence/stopping criteria were met. the inverse Hessian approximation to an identity matrix const,dat - the datafile of constants elself clambda <> pea,... dlsplay('ERROR: lambda must contain pea entries!'); retf; The function RESTART in file RESTART.MTK is used to if clambda-1, lambdanew-lambdanew.ones(1,pea); ... [ska\_opt,ske\_opt] =restart(lambdanew,stop,newh) ska opt - the optimized airframe controller ske opt - the optimized engine controller // if stop does not exist, load it from par.dat
if exist('stop')-0, load 'par.dat' stop Some parameters may be altered if desired. changelam-norm(lambda-lambdanew); // from the parameter vector p define 'partitio.mtx'; define 'linesrch.mtx'; define 'converge.mtx'; // and their sum, as
// parameter vector
define 'cost.mtx' inter.dat - the // for optimization define 'parmat.mtx' define 'parvec.mtx' lambda-lambdanew; define 'zero.mtx' Output: 22222222222

// If the difference between old lambda and new lambda is large enough, // >le-10, then call COST to initialize costs and gradients, otherwise // just use old data

abs (changelam) >1e-10, ... [grad1,grad0,grad1,fh,jh0,jh1]=COST(p\_1,lambda);... if abs(changelam)>le-10,

(cnt,tmp]=s1ze(fh); fh(cnt)=jh0(cnt)+jh1(cnt); gradi=grad0+grad1;

// If newh is defined, then set the inverse Hessian to an identity matrix, // If any of the A parameters are at the bounds (noted in fx) // then set that element of hi to be zero (no further decrease in that if l=exist('newh'), [row.col]=size(p\_l); hl=eye(row);
for l=l:atot, lf fx(l)=l, hl(i,l)=0; end,end,... // direction/parameter)

// save the possibly new stopping vector and save intermediate data save 'par.dat' stop;
save 'inter.dat' p\_1 fh jh0 jh1 grad1 grad0 grad1 h1 fx lambda;

// call the PARTITION routine which eventually returns the final
// optimized vector
p\_opt-PARTITION(stop);

Generate the optimized ska and ske as the final output of the program [ska\_opt, ske\_opt]=MAT(p\_opt);
load 'inter.dat' p\_1 fh jh0 jh1 grad1 grad0 grad1 h1 fx lambda;
// If a dummy feedback for ya or ye was added, then remove it if lya=0, [r,c]=size(ska\_opt); ska\_opt=ska\_opt(:,1:c-1); end;
if lye=0, [r,c]=size(ske\_opt); ske\_opt=ske\_opt(:,1:c-1); end;

save 'inter.dat' p\_1 ih jh0 jh1 gradi grad0 grad1 hi fx lambda ska\_opt ske\_opt;

// INTER.DAT now contains the final results including the optimal // subcontrollers.

// Created by Phil Schmidt and students Nader Kamrani and Brian Holawecky at // The University of Akron with the support of NASA Lewis Research Center

Input:

22222222222222222

```
// Checking the input data to see if it is consistant.

if exist ('s ka') = 0, display('ERROR: s ka is missing from init.dat'); retf;

if exist ('ns ka') = 0, display('ERROR: ns ka is missing from init.dat'); retf;

if exist ('ns ke') = 0, display('ERROR: ns ke is missing from init.dat'); retf;

if exist ('ns ke') = 0, display('ERROR: ns ke is missing from init.dat'); retf;

if exist ('nsc') = 0, display('ERROR: nsc is missing from init.dat'); retf;

if exist ('np') = 0, display('ERROR: nsc is missing from init.dat'); retf;

if exist ('np') = 0, display('ERROR: nsc is missing from init.dat'); retf;

if exist ('rq') = 0, display('ERROR: nsc is missing from init.dat'); retf;

if exist ('rq') = 0, display('ERROR: from init.dat'); retf;

if exist ('la') = 0, lya = 0; else lya = 1s;

if exist ('la') = 0, lya = 0; else lya = 1s;

if exist ('la') = 0, lya = 0; else lya = 1s;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   if c<>nwo+ka_tmp+ke_tmp, ...
display('ERROR: number of columns in swo not consistant with other data');...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         display ('ERROR: number of rows in swi is not consistant with other data');...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           display ('ERROR: min. observation point frq (1,1) must be less than the ');...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               // The input and output weighting matrices swi and swo of orders nwi and nwo
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      display ('ERROR: number of columns in sp not consistant with other data');...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         display('ERROR: number of rows in sc is not consistant with other data');
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          display('ERROR: number of observation points frq(3,1) must be positive');
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          If c < nwi, display ('ERROR: number of columns in swi is less than nwi');
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              if r<>np+ma tmp+me tmp+pea+lya+lye, ...
display('ERROR: number of rows in sp not consistant with other data');
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   [f mod(frq(3,1),2)=0, ...
display('ERROR: number of observation points frq(3,1) must be odd');
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          if c<>nsc+ma tmp+me tmp+lya+lye, ...
display('ERROR: number of columns in sc not consistant with other
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       [r,c]=slze(s ka); ka_tmp = r-ns ka-pea; ma_tmp = c-ns_ka-lya;
[r,c]=slze(s_ke); ke_tmp = r-ns_ke; me_tmp = c-ns_ke-pea-lye;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                display('max, observation point frq(2,1)'); retf; end;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         Get row and column of the centralized controller sc
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          if exist('sw'') =0, nwi=1; swi=eye(1+csc-nsc); end
if exist('swo') =0, nwo=1; swo=eye(1+csc-nsc); end
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               exist ('swo')=0, nwo=1; swo=eye (1+rsc-nsc); end
                             exist('sp')=0, if exist('spaug')=1, sp=spaug;
exist('np')=0, if exist('nspaug')=1, np=nspaug;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    is not
is not
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        be identity matrices of the desired size
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    if exist('nwi')=0, display('ERROR: nwi
if exist('nwo')=0, display('ERROR: nwo
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               1f r<>nw1+ma_tmp+me_tmp+lya+lye, ...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         the centralized controller)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          1f frq(1,1) >= frq(2,1), ...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               1f r<>nsc+ka_tmp+ke_tmp,
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   1f c<>np+ka tmp+ke tmp,
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   if mod(frq(3,1),2)-0,
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                1f frq(3,1) <= 0, ...
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         [rac, csc] -s [ze (sc);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          [r, c] =size(swi);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           (r,c)=size(swo);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    [r, c] =size (sb);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          (r, c) =size (sc);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    retf; end;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     retf; end;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              retf; end;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            retf; end;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              retf; end;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    retf; end;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     retf; end;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     // Specify the format for displaying matrices
// Initializing answers in case of an abrupt error/end
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   Define UDF PARTITION which is the outermost loop and implements the Broyden-Fletcher-Goldfarb-Shanno technique for finding a search direction
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        eta = tolerance for max change in norm of gradient giving convergence
                                                                                                                                                                                                                                                                                                                           lambda - a scalar or pea x 1 vector of weightings for determining the relative contribution of fl(p), the tracking cost, to the total cost.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  // Define UDF INACCURATE which implements Fletchers inaccurate line search
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     // Define the UDF MAT which generates the SKA and SKE collections of the
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               Define UDF MODI, which puts a system matrix into a modified modal form
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           // Define UDF RESTART which can restart the program using checkpointed
                                                                                                                  controller partitioning optimization code.
It sets up initial data and precalculates some required information,
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             and their sum, as well as the respective gradients for a specified
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          eps = tolerance for max change in parameters giving convergence
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  fmin = minimum tolerance for the total cost; a cost below this
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        // Define UDF LONGCOL which generates the parameter vector from the
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        // Define UDF COST which evaluates the performance cost, tracking
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            Define UDF CONVERGE which checks if convergence conditions are

    tolerance for max change in cost giving convergence

                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         // System stack size is increased
// Initial data is loaded
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           INIT.DAT - the data file containing the initial information
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              Define UDF 2 which returns a zero matrix of the desired size
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 iter - number of iterations to run; defaults to 100
                                                                                                                                                                                                                                                                                                                                                                                                            stop - (optional) the vector of stopping conditions
                                                                                                                                                                                                        calls COST to initialize the costs and gradients and calls PARTITION to do the optimization.
                                                                                 The function START in file START.MTX initiates the
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  ska opt - the optimized airframe subcontroller ske opt - the optimized engine subcontoller
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 default: stop=[le-7;le-7;le-7;le2;le-2];
                                                                                                                                                                                                                                                                                                                                                                                                                                                         stop=[eps;delta;eta;iter;fmin] where
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             parameters from the parameter vector
[ska_opt, ske_opt]=start(lambda,stop)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  is assumed to be a minimum.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        SKA and SKE system controllers.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                resize('sstack', 1000000);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          // Define UDF PARTITION
// Broyden-Fletcher-Gold
// for optimization
define 'partitio.mtx';
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          define 'linesrch.mtx';
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                define 'converge.mtx';
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            ska_opt=0; ske_opt=0;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        define 'restart.mtx';
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    parameter vector
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     define 'parmat.mtx'
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            // SKA and SKE syst
Define 'parvec,mtx'
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   define 'modl.mtx';
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  define 'zero.mtx'
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         load 'init.dat';
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   // information
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          delta
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        Output:
```

:

defined in init.dat');retf;
defined in init.dat');retf;

(depending on the size of

[ke metot] size (de); kpmpea-kptot-pea+1; If exist ('le') =0, splut "kptot -pea; me-metot-pea-le lye-le;... nk-nsc; ka-katot-pea; ma-matot-la; end, ... lya-la; sk=sc; if rlambda > 1, disp('ERROR: lambda must be a row vector or a scalar!'), retf;
if clambda-1, lambda-lambda\*ones(l,pea), ... centralized controller. Under option 1 DEEA and DEAA are determined by optimization. Under option 2 DEEA is set to 0 and DEAA is determined inquire sore 'ENTER 1 TO FIX AIRFRAME, 2 TO FIX ENGINE or 0 FOR NEITHER: '; optimization. Under option 3 DEEA-0 and the remaining Ds are determined // Get the row and column size of the plant matrix, input weighting matrix, // increase the sizes of the system matrices to accomodate these
// feedback variables. In the case of the dummy feedbacks being created
// (and not input via init.dat) the system matrices are adjusted with
// zero columns.
if exist('ia')-0, Inquire if the airframe or the engine submatrices are to be held fixed DISPLAY ('ENTER 1 TO FIX ALL DS EXCEPT DEAA & DEEA, 2 TO INCLUDE DEEA,'); ş if aore=0, atot=nka+nke; ska=modl(s\_ka,ns\_ka); ske=modl(s\_ke,ns\_ke);...
elself aore=1, atot=nka; ska=s\_ka; ske=modl(s\_ke,ns\_ke);...
elself aore=2, atot=nka; ska=modl(s\_ka,ns\_ka); ske=s\_ke;...
else DISPLAY('YOU MUST ENTER 0, l or 2('); retf;... // Inquire if any of the D submatrices are to be fixed. DAA, DAYA, // DEE, and DEYE are determined by the corresponding submatrices of the if r < nwo, display('ERROR: number of rows in swo is less than nwo');</pre> // The stopping conditions vector is given a default value if it was
// provided (it will be saved in par.dat) // Inquire if the airframe or the engine submatrices are to be held f
// storing the reply in variable aore
// If not, put the subcontroller matrices in modified modal form.
// Hake ska and ske with orders nka and nke the variables to be used
// from now on (instead of s\_ka and s\_ke) // If the measurements from plant to controllers la,le, are absent, // create a dummy feedback of one (la = l and/or le = l). // Verify that the lamda provided is a row vector or a scalar and // lya and lye hold the actual number of ya and ye measurements. display ('ERROR: lambda must contain pea entries!'); retf; 3 FOR ONLY DEEA, or 0 for NONE:'; by optimization. Under option 0 no Ds are fixed. DISPLAY ('YOU MUST ENTER 0, 1, 2 or 3!'); retf;and, // and the airframe and engine controller matrices. if exist('stop')=0, stop=(le-7;le-7;le-7;le2;le-2]; if (fixd-1)\*(flxd-2)\*(flxd-3)\*fixd <> 0,... swi=[swi; 0\*ones(2,cswi)];... ske=[ske, 0\*ones(rske,1)];... ska=[ska, 0\*ones(rska,1)];... sc=[sc, 0\*ones(rsc,2)];... sp=(sp; 0\*ones(2,csp)];... [rlambda, clambda] - size (lambda); elseif clambda <> pea,... if exist ('le')"0, .... [rswi, cswi] =size(swi); nke-ns ke; [rska, cska] "size (ska); [rske, cske] =size(ske); consistant with pea (rsp, csp) -size(sp); save 'par.dat' stop la-1; lya-0; ... le=1; lye=0; ... else lyemle;... inquire fixd' nka-ns ka; retf: end:

<u>د</u>1



// to get ma, the actual number of inputs to ska. Subtract the number // of pea airframe to engine commands from the katot to get ka the number // If any of the entries of the first column of ba or be are <= zero (or sw!=[sw!([l:rsw!-le],:); O'ones(l,csw!); sw!([rsw!-le+1:rsw!],:)];... commands, and le, the bal-ba(:,1); bel-be(:,1); // This is required by the modified modal form of the system matrix. to a matrix of ones // ska=[aa,ba;ca,da]. Use da to figure out the total inputs matot // and outputs katot for ska. Subtract the la input from matot // almost <= 0) then set them to le-9 (almost zero but positive). sc=[sc(:,[1:csc-le]),0\*ones(rsc,1),sc(:,[ssc-le+1:csc])];... sp=[sp([1:rsp-le],:); 0\*ones(1,csp); sp([rsp-le+1:rsp],:)];... // Decompose the airframe subcontroller in modified modal form Decompose the engine subcontroller in modified modal form matrix Nperf for 1-1:nka, if abs(ba(1,1))<1e-9, ba(1,1)=1e-9; end,end, for 1-1:nke, if abs (be(1,1))<1e-9, be(1,1)=1e-9; end,end, controller sk=(ak,bk;ck,dk] // Decompose the integrated plant sp=[ap,bp;cp,dp] Nperf = max(Nperf,le-13); [r,c]=slze(Nperf);... // normalizing the performance cost, it is set // This matrix should be a frq(3,1)x1 matrix. // If the user has not entered the (optional) // Let sk-sc be the centralized controller ske=[ske, 0\*ones(rske,1)];... swi=[swi; O\*ones(l, cswi)];... // of outputs going to the plant sp={sp; 0\*ones(1,csp)}; ... sc=[sc, 0\*ones(rsc,1)];... // Decompose the centralized [aa ba ca da] -split (ska, nka); [ae be ce de]-split (ske, nke); (ak,bk,ck,dk)=split (sk,nk); (ap, bp, cp, dp] =split (sp, np); 1f exist('Nperf')=1, ... [katot matot] =size(da); [kptot ntmp] -size (cp); [ntmp mptot] = size (bp); lf r<>frq(3,1),... le=1; lye=0; ...

# start.mtx

if aore<>1,if (fixd-1) \* (fixd-2) \*\*0, da\*\*(dak, dyak; da ((ka+1:ka+pea],:)); end; end; Generate the optimized ska and ske as the final output of the program omega daa dee daya deye atot sg ng swi nwi swo nwo weight stabil ... bal bel ska0 ske0 aore Nperf Ntrack fixd; // call the PARTITION routine returning the final optimized vector Store the current (Initial) values of ska and ske in skaO and If it is not defined or it is defined too high, it is given a // Let sg be the constructed state space representation of Tcent save 'start.dat' aore p\_1 grad1 grad0 grad1 fh jh0 jh1 lambda if (fixd-1) \* (fixd-2) = 0, de = (de(:, [1:pea]), dek, dyek]; end:... // If Ds are fixed, their initial values are determined here be read in from init.dat and later stored in const.dat The stability upperbound stabil for the A submatrices dee=de(:, {pea+1:pea+me}); deye=de(:, [pea+me+1:metot]); daa-da((1:ka),(1:ma)); daya-da((1:ka),[ma+1:ma+la]); (grad1, grad0, grad1, fh, jh0, jh1)-cost (p\_1, lambda); to be used if the subcontroller(s) are fixed save 'const.dat' la le lya lye ka ma ke me pea // if there was no error in the data checking ska={aa, ba; ca, da]; ske={ae,be; ce, de]; save 'inter.dat' p\_l grad! grad0 grad1 exist ('stabil') =0 then stabil==1e-9; 1f aore <> 2, 1f (flxd-2)\*(flxd-3)=0, Save starting data for restarts Save the constants in const.dat // Initialize costs and gradients stabil>-le-9, stabil --le-9; [ska\_opt, ska\_opt] "MAT (p\_opt); Save intermediate results ceab : p\_opt-PARTITION (stop) ; p 1-longcol (ska, ske); [row, col] -size(p\_1); for I=1:atot, ....
If p\_1(I)>stabil,
p\_1(I)=stabil; bg- (bak; bpa-dak); cg= (0\*ones (pea, nk) dq= 0 ones (pea, ma); ska0-ska; ske0-ske; fx=0\*ones (atot, 1); h1 (I, I) =0; ... (x(I)=1; ... sg=[ag, bg;cg, dg]; default value hi-eye (row); ng= nk+np3 // Define the set of frequency points at which the cost function and gradient // are evaluated. Store as the variable OMEGA.. omega-frq(1)\* $\{(frq(2)/frq(1))^**\{1/\{frq(3)-1\})\}^*$ S display ('ERROR: The number of performance weights is different from the ... byak= bk(:, [(ma+me+1): (ma+me+1a)]); byek=bk(:,[(ma+me+1a+1):(ma+me+1a+1e)]);
cak= ck(1:ka,:); cek= ck(kap1:kc,:); ۵ HERE Construct the state space representation of Tcent from input: Zac // The centralized controller is decomposed in a similar way as the plant // Define the weights for Simpsons rule integration. They depend on the  $\prime\prime$  number of observation points specified in frq(3) and are stored in Decompose the plant controller, splitting up the matrix bp relative deak-dk([ka+1:ka+ke],[1:ma]);deyak-dk([ka+1:ka+ke],[ma+me+1:ma+me+la]); // its inputs and the matrix cp relative to its outputs. For example, // cap would be the plant to airframe place, the first ma rows of the // plant matrix because these rows form the input to the airframe. [mp ntmp]-size(cp); // output: zea using the system with the global controller. This // is used in evaluating the zea tracking cost, but does not alter // with the choice of parameters so it is constructed once only // in this routine and then stored.

// in this routine and then stored.

gg {ak, (-bak\*cap-bek\*cep+byak\*cyap+byek\*cyep);... display ('ERROR: The number of tracking weights disagrees with the display ('ERROR: The number of tracking weights disagrees with the bpa\*dyak\*cyap+bpe\*dyek\*cyep-bpa\*daek\*cep+bpa\*dayek\*cyep... // If the user has not entered the (optional) matrix Mtrack for // normalizing the tracking cost, it is set to a matrix of ones.
// This matrix should be a frq(3,1) xpea matrix.
if exist ('Ntrack')-1, ... (bpa\*cak+ bpa\*cek), (ap-(bpa\*dak\*cap)-(bpe\*dek\*cep)+... dak- dk([1:ka], [1:ma]); daek-dk([1:ka], [ma+1:ma+me]); Ntrack = max(Ntrack,le-13); [r,c]=size(Ntrack);... dyek=dk((ka+1:ka+ke), (ma+la+me+l):(ma+la+me+le)); cap = cp((1:ma),:); ceap = cp(((ma+1):(ma+pea)],:); cyep~cp(((ma+me+pea+la+1):(ma+me+pea+la+le)],:); for 1-1: ((frq(3,1)-3)/2), weight-[weight, 4,2]; end number of interface variables '); retf; end;... // controller [kc ktmp]=size(ck); bak= bk(:,1:ma); bek= bk(:,[(ma+1):(me+ma)]); dayek dk([1:ka], [ma+me+la+la+le]); number of frequency points '); retf; end; ... number of frequency points ');retf;end;... cyap=cp([ (ma+me+pea+1) : (ma+me+pea+1a) ], :); bpa- bp(:,[1:ka]); bpe- bp(:,[(ka+1):kp]); -bpe deak cap bpe deyak cyap) ]; dyak=dk({1:ka}, (ma+me+1:ma+me+la]); dek=dk([ka+1:ka+ke],[ma+1:ma+me]); else, Ntrack-ones (frq (3,1), pea);... cep- cp((ma+pea+1): (ma+pea+me),:); else, Nperf-ones(frq(3,1),1);... // the variable WEIGHT. kp- ka+ke; kapl- ka+l; 1f r<>frq(3,1),... welght-[welght, 4, 1]; c<>pea,...

welght-[1];

ť

hi lambda

th Jho Jhl



de(:, [1:pea]) =0\*ones(ke,pea); end;...

ske0

save 'inter.dat' p\_i fh jh0 jh1 gradi grad0 grad1 hi fx lambda ska\_opt ske\_opt; hi fx lambda; // If a dummy feedback for ya or ye was added, then remove it if lya-0, [r,c]-size(ska\_opt); ska\_opt-ska\_opt(:,l:c-l); end; if lye-0, [r,c]-size(ske\_opt); ska\_opt-ske\_opt(:,l:c-l); end; load 'inter,dat' p i fh jh0 jh1 grad1 grad0 grad1 save final data

inserts dummy values of 1 and loads appropriate zeros in the data matrices.) stabil - upper bound for eigenvalues of Aa and Ae to assure stability is optional weighting matrices swi and swo (for input and/or output weighting) controllers s ka and s ke, and the plant sp along with their orders nsc, ns ke, and np resp. It also contains the number of intermediate variables pea and the range of frequencies for calculating norms frq. The may be included (default - Id in both cases) as may la and/or le, the numbers of neg. feedback variables (if these are absent, then the code program. It expects the input to the program to be in the datafile 'init.dat', and requires certain data to be there. 'init.dat' contains the global controller sc, the initial partitioned The UDF start.mtx is the main routine for the controller partitioning also contained in init.dat (if missing, defaults to -le-9). ::::::::::

The code gives the option of fixing values of either subcontroller if // optimization over the parameters in only one subcontroller is desired

The option is also given to fix certain of the D submatrices.

// consist of 2x2 companion blocks and the first column of the B matrix is // fixed to its initial value (or le-9 if this initial value is zero). These // are then split to identify the state space representations from inputs to The subcontrollers are put into a modified modal form where the A matrices

Here the modifications are made to account for the absence of la and/or le. outputs using intermediate variables.

The logarithmic frequency range over which the cost function is evaluated is created as Omega. The vector of Weights used in Simpson's Rule integration is also calculated.

transfer matrix Tcent: zac to zea is computed, sg. Its row norms are stored be used as normalizations in the frrack part of the cost function. The state-space representation for the nominal "command-tracking" ç

values of ska0 and ske0 which are used in case the appropriate subcontroller is fixed, atot indicates the sum of orders of the subcontroller state-space numbers, i.e. 0 if la resp le were absent in 'init.dat'. In this case la/le the numbers of negative feedbacks from the plant (lya/lye relects the actual the vector of row norms for the nominal tracking transfer matrix. Other constants are the values of the D subcontroller submatrices which are fixed nominal tracking transfer matrix, sg; the frequency weighting matrices, swi and swo; and their orders --- np, nk, ng, and nw along with the orders vector of frequencies, omega; the weights for Simpson's Rule, weight; and equal to the corresponding submatrices of the state space representation for the centralized controller; the first columns of the subcontroller B would contain the dummy value 1). Further constants are the state space The file 'const.dat' contains constants --- the numbers of subcontroller inputs, ma and me; outputs, ka and ke; and intermediate variables, pea; of the subcontrollers nka and nke; plus the frequency range, frq; the matrices which are fixed by the modi canonical form; and finally, the matrices for the plant, sp; the centralized controller, sk; the \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

into the form of a parameter vector p\_1, an identity matrix is created as The initial system matrices for the subcontrollers ska0 and ske0 are put

matrices which are not fixed.

// process and a vector, fx, indicating which of the entries in the state-space // A matrices are at the stability upper bound. p\_i and lambda are passed to // the initial approximation for the inverse Hessian used in the optimization // the COST function which returns the initial values of:

jho - the performance cost fh = the total cost

1h1 - the tracking cost

gradi = gradient of the total cost grad0 = gradient of the performance cost

grad! - gradient of the tracking cost

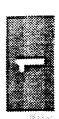
These values are stored in 'inter.dat' the file for intermediate results. 'start.dat' contains a record of the starting configuration. The parameters are passed to PARTITION which carries out the optimization. This procedure reads the data it needs from the file 'inter.dat' and returns the optimized paramter vector p\_opt.

This final vector is split into the optimized airframe and

controllers ska opt and ske opt, which are the final output of

Created by Phil Schmidt and students Nader Kamrani and Brian Holawecky at The University of Akron with the support of NASA Lewis Research Center under grant NAG-3-1146.

63



```
//czer> = z(nrow,ncol)
// Construct a matrix of zeros of size nrow x ncol. This is done so that
// less space is taken within the main program
// Input:
// nrow - number of rows
// ncol - number of columns
// output:
// zer - the zero matrix of the desired size (nrow,ncol)
                                                                                                                                                                                                                                                                                                                                                                                                        // Created by Phil Schmidt at The University of Akron with the support // of NASA Lewis Research Center under grant NAG-3-1146.
                                                                                                                                                                                                                                                                                            zer=0*ones (nrow, ncol);
                                                                                                                                                                                                                                                                                                                                                                 retf
```

APPENDIX IV

DATA FOR EXAMPLE

The data file INIT.DAT corresponding to the example contains the following:

PEA = 1 — number of interface variables

FRQ = [0.1; 100; 41] — describes the frequency range of 41 points between 0.1 and 100

NP = 13 - order of the integrated plant

NSC = 13 - order of the centralized controller

NS\_KA = 10 — order of the airframe subcontroller

NS\_KE = 7 — order of the engine subcontroller

NWI = 26 - order of the input weighting transfer matrix

NTRACK =

```
6.1823D - 01 6.3153D - 01 6.4195D - 01 6.5131D - 01 6.6139D - 01
                                                                            6.7411D - 01 · · ·
                                                                            8.7492D - 01 \cdots
                                            7.5413D - 01
                                                            8.0582D - 01
                              7.1732D - 01
                                                                            1.1194D + 00 \cdots
                                                           1.1646D + 00
                             1.0534D + 00
                                             1.1326D + 00
               9.6037D - 01
                                                                            4.9513D - 01 · · ·
                                                            5.6432D - 01
                                             6.8055D - 01
                              8.3547D - 01
               9.9744D - 01
                                                                            4.6188D - 01 \cdots
                                             4.5255D - 01
                                                            4.5595D - 01
                              4.5302D - 01
               4.6326D - 01
                                                                            8.9018D - 01 \cdots
                                                            6.6494D - 01
                                             5.5002D - 01
                              4.9653D - 01
               4.7282D - 01
                                             3.0733D + 00 4.8371D + 00
                                                                            7.6375D + 00 \cdots
               1.2933D + 00 \quad 1.9711D + 00
                                                                           7.5645D + 01
               1.2067D + 01 1.9074D + 01 3.0171D + 01 4.7762D + 01
```

— 41  $\times$  1 vector of tracking weights (The size of NTRACK is FRQ(3) $\times$  PEA.)

SP (listed below) — state-space matrix for the integrated plant transfer matrix

SC (listed below) — state-space matrix for the centralized controller transfer matrix

S\_KA (listed below) - state-space matrix for the initial airframe subcontroller transfer matrix

S\_KE (listed below) — state-space matrix for the initial engine subcontroller transfer matrix

SWI (described below) — state-space matrix for the input weighting transfer matrix transfer matrix

The data matrices and initial partitioning matrices for the controller partitioning example are listed below. In all cases the given matrices A, B, C and D correspond to the state-space representation of the given system or subsystem

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = A\mathbf{x} + B\mathbf{y}$$
$$\mathbf{u} = C\mathbf{x} + D\mathbf{y}.$$

The state-space matrix for this system is S where  $S = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ .

The integrated airframe propulsion system with integrator augmentation is represented by  $SP = \begin{bmatrix} AP & BP \\ CP & DP \end{bmatrix}$  where  $3.81E^{-2}$  $3.14E^{-4}$  $2.59E^{-4}$  $1.40E^{-2}$  $-3.85E^{+1}$  $-3.18E^{+1}$  $3.60E^{-2}$  $-4.40E^{-2}$  $-2.10E^{-6}$  $1.82E^{-4}$  $-1.57E^{-5}$  $5.19E^{-4}$  $-4.59E^{+0}$  $-4.46E^{-1}$  $1.94E^{+2}$  $-2.27E^{-1}$  $3.74E^{-7}$  $3.66E^{-5}$  $2.56E^{-5}$  $9.46E^{-7}$  $-4.81E^{-4}$  $1.51E^{-2}$  $-1.94E^{-1}$  $-3.09E^{-3}$  $1.00E^{+0}$  $2.00E^{+2}$  $-9.89E^{-1}$  $1.42E^{-1}$  $-3.43E^{+2}$  $-4.19E^{+0}$  $6.02E^{+0}$  $-8.48E^{-2}$  $1.54E^{-1}$  $7.78E^{-1}$  $2.71E^{+1}$  $-5.70E^{+0}$  $4.26E^{-1}$  $-1.65E^{-2}$  $3.00E^{-2}$  $1.51E^{-1}$ AP = $-9.02E^{+1}$  $2.29E^{-1}$  $1.15E^{-1}$  $-3.50E^{-1}$  $1.57E^{-1}$  $7.93E^{-1}$  $-1.03E^{-1}$  $3.74E^{-2}$  $-7.95E^{+0}$  $1.09E^{-2}$  $-1.00E^{-1}$  $-1.99E^{-2}$  $6.91E^{+0}$  $-1.02E^{+1}$  $1.71E^{-1}$  $2.25E^{-3}$  $-5.45E^{+0}$  $6.17E^{-5}$  $-1.46E^{-2}$  $7.10E^{-3}$  $-2.95E^{-6}$  $-8.38E^{-3}$  $-7.97E^{-1}$  $5.33E^{-3}$  $2.67E^{-6}$  $2.75E^{-4}$  $7.34E^{+2}$  $1.16E^{+1}$  $2.68E^{+2}$ ...  $1.03E^{+1}$  $-2.15E^{+3}$  $-2.58E^{+3}$  $8.47E^{-1}$  $9.06E^{+1}$  $-1.06E^{+0}$  $8.21E^{+2}$  $-1.00E^{-2}$  $-1.00E^{-2}$  $-1.00E^{-2}$  $-1.00E^{-2}$ BP = $1.00E^{+0}$  $1.00E^{+0}$  $1.00E^{+0}$  $1.00E^{+0}$  $\Gamma 4.89E^{-2}$   $9.70E^{-3}$  $1.71E^{+0}$  $1.71E^{+1}$  $1.74E^{-3}$ CP = $2.04E^{-1}$  $3.15E^{-3}$  $1.20E^{-3}$  $3.45E^{-5}$  $2.31E^{-5}$  $2.40E^{-5}$ 

 $1.92E^{-4}$ 

 $1.47E^{-3}$ 

 $5.68E^{-1}$ 

 $-8.53E^{-1}$ 

The augmented plant state variables are

$$\mathbf{x} = [u, w, q, \theta, h, N2R, N25, P6, T41B, \delta_{TV}, WF, A78, A8]$$

where

u = aircrast body axis forward velocity, ft/s

w = aircraft body axis vertical velocity, ft/s

q = aircraft pitch rate, rad/s

 $\theta = \text{pitch attitude, deg}$ 

h = altitude, ft

N2R = engine fan speed, rpm

N25 = core compressor speed, rpm

P6 = engine mixing plane pressure, psia

T41B = engine high pressure turbine blade temperature, ° R

 $\delta_{\rm TV} = {\rm normalized}$  thrust vectoring angle, deg/10

WF = normalized engine main burner fuel flow rate, lb m/hr/500

 $A78 = \text{normalized thrust reverser port area, in}^2/50$ 

 $A8 = \text{normalized main nozzle throat area, in}^2/100.$ 

The augmented plant outputs are listed in the order  $\begin{bmatrix} \mathbf{z}_a \\ \mathbf{z}_e \\ \mathbf{z}_{ea} \end{bmatrix}$  with  $\mathbf{z}_a = \begin{bmatrix} V \\ q_v \end{bmatrix}$ ,  $\mathbf{z}_e = \begin{bmatrix} N2 \\ EPR \end{bmatrix}$ , and  $\mathbf{z}_{ea} = FEX$  where

V = normalized aircraft speed, ft/s/20

 $q_v = \text{normalized pitch variable, } (q(\text{deg/s}) + 0.1\theta(\text{deg}))/3$ 

N2 = normalized engine fan speed, % of maximum allowable rpm at operating condition/5

EPR = normalized engine pressure ratio, ratio/0.3

The augmented plant inputs are the rates of change on normalized control variables as described in the text.

```
The centralized controller has state-space representation SC = \begin{bmatrix} AC & BC \\ CC & DC \end{bmatrix} where
```

```
1.40E^{-2} -4.41E^{-4}
                                                                                       2.59E^{-4} -5.38E^{-2}
                                   3.63E^{+1} -2.43E^{+1}
                     -2.00E^{-1}
        -1.23E^{+0}
                                                                                     -2.10E^{-6}
                                                                                                    1.21E^{-3}
                                                             5.19E^{-4}
                                                                          1.59E^{-5}
                     -4.07E^{-1} -5.53E^{+1}
                                              -2.96E^{+1}
        -3.21E^{-2}
                                                                                       3.74E^{-7} -8.20E^{-6}
                                                             2.56E^{-5}
                                                                          1.08E^{-6}
                      1.52E^{-2} -6.49E^{+0} -6.30E^{-1}
        -2.89E^{-3}
                                                                                                    4.50E^{-5}
                                                                        -1.80E^{-7}
                      7.43E^{-6} -1.34E^{-2} -1.01E^{-1}
                                                                                         0
                                                              0
         3.75E^{-5}
                                                                                                    3.17E^{-3}
                                                                          1.05E^{-4}
                                                                                         0
                                                2.02E^{+2}
                                    2.02E^{+1}
                                                              0
        -4.36E^{-1}
                     -1.10E^{+0}
                                                                                       6.02E^{+0}
                                                                                                  -2.20E^{+2}
                                                                        -7.78E^{+0}
                                                1.14E^{+0} -8.48E^{-2}
        1.86E^{-1}
                       3.70E^{-2}
                                    1.14E^{+1}
                                                1.07E^{-1} -1.65E^{-2} -9.61E^{-1}
                                                                                     -5.70E^{+0} -2.47E^{+1} ...
                                   1.07E^{+0}
                     -3.53E^{-2}
AC = |-1.78E^{-1}|
                                                                                      1.15E^{-1} -9.87E^{+1}
                                                                          2.38E^{-1}
                                                           -3.50E^{-1}
                      1.56E^{-1} -2.86E^{-1} -2.86E^{-2}
       7.88E^{-1}
                                                                                    -1.03E^{-1} -2.86E^{+1}
                                                             1.09E^{-2} -7.49E^{-1}
                     -4.88E^{-2}
                                  -1.26E^{+0} -1.26E^{-1}
        -2.46E^{-1}
                                               1.73E^{+2}
                                                             1.79E^{-3}
                                                                         2.70E^{-5}
                                                                                      8.48E^{-6} -8.94E^{-4}
                                  1.73E^{+3}
                      1.00E^{+0}
        -1.32E^{-1}
                                                                                    -6.16E^{-2} -2.72E^{-1}
                                                           1.76E^{-3} -1.09E^{-1}
                                                7.47E^{-2}
                    -7.60E^{-3}
                                  -4.04E^{-1}
        -3.77E^{-2}
                      7.08E^{-1} -2.26E^{+1} -2.41E^{+1} -9.57E^{-3} -1.18E^{-2} -1.24E^{-3} 8.23E^{+0}
        3.80E^{+0}
                                                 2.00E^{+1} -3.12E^{-2} -1.39E^{-2} -1.29E^{-3} 9.88E^{+0}
       L - 3.06E^{+0}
                     -5.68E^{-1}
                                    1.96E^{+1}
                                                             2.25E^{-3} -4.18E^{-3}
                                                                                       1.71E^{-1} -1.02E^{+1}
                                                                                                                 6.91E^{+0}
                                                                                                                 7.10E^{-3}
                                                                                       6.17E^{-5} -1.46E^{-2}
                                                            -2.95E^{-6} -5.45E^{+0}
                                                                                       2.75E^{-4}
                                                                                                    5.33E^{-3} -8.38E^{-3}
                                                              2.67E^{-6} - 7.97E^{-1}
                                                                                          0
                                                                                                       0
                                                                                                                    0
                                                                             0
                                                                                                                    0
                                                                                                       0
                                                                             0
                                                                                          0
                                                                0
                                                                                       7.34E^{+2}
                                                                                                                    0
                                                              1.16E^{+1}
                                                                                                       0
                                                                             0
                                                                                       2.68E^{+2}
                                                        ... 1.03E^{+1}
                                                                                                                    0
                                                                             0
                                                                                                      0
                                                                                                  -2.15E^{+3}
                                                                                                                -2.58E^{+3}
                                                                                       9.06E^{+1}
                                                             8.47E^{-1}
                                                                             0
                                                                                       8.21E^{+2}
                                                            -1.06E^{+0}
                                                                                                       0
                                                                             0
                                                                                       1.06E^{-2}
                                                                                                    4.16E^{-1}
                                                                                                               -3.58E^{-1}
                                                              9.44E^{-5}
                                                                        -5.24E^{+1}
                                                                                                                 4.31E^{+0}
                                                                                                    3.64E^{+0}
                                                                         1.06E^{-2} -2.33E^{+1}
                                                            -2.14E^{-1}
                                                                                       3.64E^{+0} -1.25E^{+2} -1.37E^{+2}
                                                                          4.16E^{-1}
                                                              2.76E^{-2}
                                                                                       4.31E^{+0} -1.37E^{+2} -1.76E^{+2}
                                                              3.27E^{-2} -3.58E^{-1}
```

$$\mathrm{BC} = \begin{bmatrix} 2.43E^{+1} & -4.36E^{+0} & 4.33E^{-1} & 4.51E^{-1} \\ -3.99E^{+0} & 1.45E^{+1} & -1.81E^{-2} & -5.07E^{-3} \\ -4.13E^{-3} & 3.66E^{-1} & -7.81E^{-5} & 2.19E^{-4} \\ -7.66E^{-4} & 5.89E^{-2} & 1.03E^{-4} & -2.20E^{-4} \\ 1.18E^{+1} & -1.17E^{+0} & -6.02E^{-2} & -1.55E^{-2} \\ 1.20E^{+1} & -6.68E^{-1} & 2.05E^{+3} & -6.00E^{+2} \\ 6.74E^{+0} & -6.23E^{-2} & 7.95E^{+2} & 2.54E^{+2} \\ 1.08E^{-1} & 1.66E^{-2} & -5.12E^{+0} & 4.16E^{+1} \\ 2.97E^{+0} & 7.34E^{-2} & 4.50E^{+2} & 1.01E^{+2} \\ -1.55E^{-1} & -1.24E^{+0} & 1.74E^{-3} & 9.36E^{-4} \\ 1.05E^{-2} & 6.63E^{-4} & 1.16E^{+0} & 4.47E^{-1} \\ -9.50E^{-1} & 1.18E^{-1} & 2.94E^{-1} & -7.46E^{-1} \\ 7.96E^{-1} & -9.98E^{-2} & 3.36E^{-1} & -8.97E^{-1} \end{bmatrix}$$

The initial partitioning consisted of the airframe controller with representation  $S_{-}KA = \begin{bmatrix} A_{-}KA & B_{-}KA \\ C_{-}KA & D_{-}KA \end{bmatrix}$  of order 10 and an engine controller with representation  $S_{-}KE = \begin{bmatrix} A_{-}KE & B_{-}KE \\ C_{-}KE & D_{-}KE \end{bmatrix}$  of order 7. The initial partitioning was obtained by the procedure of reference [2]. This partitioning is not in the minimal parameter form. The submatrices in the initial partitioning are:

$$\mathbf{B}_{-}\mathbf{K}\mathbf{A} = \begin{bmatrix} 2.90E^{-1} & -2.68E^{+1} \\ 9.67E^{-2} & 9.22E^{-1} \\ 1.19E^{-1} & -3.63E^{-1} \\ 1.13E^{+3} & -1.04E^{+2} \\ -3.66E^{+1} & 1.90E^{+1} \\ 1.51E^{+0} & 1.58E^{+1} \\ -3.36E^{-1} & -6.41E^{+0} \\ -3.35E^{-1} & 2.59E^{+1} \\ 3.16E^{+0} & -6.02E^{+0} \\ 0 & 0 \end{bmatrix}$$

and

$$\mathbf{A}\_\mathbf{KE} = \begin{bmatrix} -3.50E^{+2} & 5.11E^{+1} & 0 & 0 & 0 & 0 & 0 & 0 \\ -5.30E^{+1} & -1.75E^{-2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.67E^{-1} & -1.90E^{+1} & -1.04E^{+0} & -4.51E^{-1} & 2.83E^{-1} \\ 0 & 0 & 1.73E^{+1} & -2.02E^{+1} & -1.06E^{+1} & -1.03E^{+0} & -6.57E^{-2} \\ 0 & 0 & 1.17E^{+0} & -1.06E^{+1} & -1.92E^{+1} & 1.89E^{+1} & -2.99E^{+1} \\ 0 & 0 & -5.29E^{+0} & 1.41E^{+1} & 1.03E^{+2} & -3.59E^{+2} & 2.50E^{+2} \\ 0 & 0 & -8.39E^{-1} & 5.67E^{+0} & 3.56E^{+1} & -2.46E^{+2} & -2.07E^{+1} \end{bmatrix}$$

$$\mathbf{B}_{-}\mathbf{KE} = \begin{bmatrix} -8.66E^{+1} & 0 & 0 \\ -5.94E^{-1} & 0 & 0 \\ 0 & 1.69E^{-2} & -1.83E^{+0} \\ 0 & -1.91E^{+1} & 7.38E^{-1} \\ 0 & -5.75E^{+0} & 5.45E^{+0} \\ 0 & 5.49E^{+0} & -2.93E^{+1} \\ 0 & 2.51E^{+0} & -4.48E^{+0} \end{bmatrix} \qquad \mathbf{D}_{-}\mathbf{KE} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\text{C.KE} = \begin{bmatrix} -4.12E^{-1} & -1.44E^{-1} & -1.81E^{+0} & -1.91E^{+1} & -5.61E^{+0} & -1.45E^{+0} & 4.77E^{-1} \\ 7.47E^{+1} & 1.45E^{-1} & 1.86E^{-1} & -6.92E^{-1} & -3.67E^{+0} & 1.91E^{+1} & -3.44E^{+0} \\ -4.37E^{+1} & 5.58E^{-1} & 2.15E^{-1} & -9.85E^{-1} & -4.22E^{+0} & 2.28E^{+1} & -3.79E^{+0} \end{bmatrix}$$

Recall that the input weighting transfer matrix is  $W_i(s) = G(s)(I+K(s)G(s))^{-1}$  where G(s) and K(s) are the integrated plant and centralized controller transfer matrices respectively. The state-space representation for this transfer matrix is too large to list. Instead, the user can easily construct it from the following MATRIXX command applied to the state space matrices for the integrated plant and the centralized controller

[SWI,NWI] = FEEDBACK(SC,NSC,SP,NP).

Notice that this command also produces the correct value for NWI, the order of the weighting transfer matrix.

The parameter optimization algorithm for controller partitioning was applied to the problem with the initial partitioning given by S\_KA and S\_KE as listed above and with input weighting SWI as described above. The controllers obtained from this process had state-space representations SKA\_OPT for the airframe and SKE\_OPT for the engine where, as before

$$\mathbf{B\_A\_OPT} = \begin{bmatrix} -3.67E^{-3} & -5.45E^{+3} \\ -1.24E^{-2} & 1.60E^{+3} \\ 1.31E^{+5} & -9.08E^{+3} \\ 8.50E^{+4} & -2.58E^{+4} \\ -4.23E^{-1} & 2.03E^{+1} \\ -6.84E^{+0} & 9.54E^{+2} \\ 1.03E^{+4} & -1.84E^{+3} \\ -2.11E^{+2} & 4.55E^{+2} \\ 2.14E^{-1} & 6.51E^{+3} \\ -7.97E^{-2} & -6.50E^{+2} \end{bmatrix}$$

$$\mathbf{D\_A\_OPT} = \begin{bmatrix} 2.82E^{-1} & -2.94E^{-3} \\ 8.98E^{-4} & -1.59E^{-3} \end{bmatrix}$$

$$\text{C\_A\_OPT} = \begin{bmatrix} -3.86E^{-5} & -1.06E^{-4} & -4.69E^{-5} & -5.57E^{-6} & -9.82E^{+0} & -5.41E^{-1} & -1.85E^{-6} \\ 2.49E^{-4} & 2.03E^{-3} & 1.91E^{-6} & -1.28E^{-5} & -2.99E^{-3} & 2.17E^{-4} & 7.65E^{-5} & \cdots \\ & & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & \\ & & & & & & & & & & & & & \\ & & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & \\ & & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & \\ & & & & & \\ & & & & & \\ & & &$$

The optimized engine controller has state-space representation

$$\mathbf{A}.\mathbf{E}.\mathbf{OPT} = \begin{bmatrix} 0 & 1.00E^{+0} & 0 & 0 & 0 & 0 & 0 & 0 \\ -2.12E^{+2} & -1.65E^{+1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.00E^{+0} & 0 & 0 & 0 & 0 \\ 0 & 0 & -7.15E^{+4} & -3.89E^{+2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.00E^{+0} & 0 \\ 0 & 0 & 0 & 0 & -1.20E^{+2} & -2.35E^{+1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -3.41E^{+2} \end{bmatrix}$$

$$\mathbf{B.E.OPT} = \begin{bmatrix} 1.00E^{-9} & -2.53E^{+1} & 5.04E^{+0} \\ 1.00E^{-9} & 9.38E^{+1} & -7.78E^{+1} \\ 1.00E^{-9} & 3.75E^{+0} & -1.56E^{+1} \\ 1.00E^{-9} & 1.67E^{+3} & -7.69E^{+3} \\ 1.32E^{+1} & -1.35E^{+1} & 2.95E^{+0} \\ -1.05E^{+2} & 1.94E^{+2} & -4.60E^{+1} \\ 8.96E^{+1} & -3.56E^{-1} & 8.92E^{-2} \end{bmatrix} \qquad \mathbf{D.E.OPT} = \begin{bmatrix} 5.82E^{-1} & 1.72E^{+0} & 1.41E^{-2} \\ -2.04E^{+0} & 1.48E^{+0} & 1.10E^{-1} \\ 6.64E^{+0} & -6.93E^{-1} & 1.03E^{-2} \end{bmatrix}$$

$$\mathbf{C}_{-}\mathbf{E}_{-}\mathbf{OPT} = \begin{bmatrix} -4.65E^{+0} & -1.07E^{+0} & -1.42E^{-1} & -9.50E^{-4} & 2.52E^{+1} & 3.36E^{+0} & 5.68E^{-1} \\ 1.97E^{+0} & 1.44E^{-1} & 8.80E^{+0} & 5.88E^{-2} & 2.14E^{+1} & 1.46E^{+0} & -7.44E^{+1} \\ -1.62E^{+0} & -1.10E^{-1} & 1.01E^{+1} & 6.27E^{-2} & -1.92E^{+1} & -1.37E^{+0} & 4.47E^{+1} \end{bmatrix}$$

Residualization of high frequency modes was applied to the optimized engine controller to reduce it to one with order 4 (not shown here because it is easily obtained). Balanced model reduction was applied to the optimized airframe controller to reduce it to one of order 6. The optimization procedure was applied to this sixth order subcontroller with the engine controller fixed at the one of fourth order. The resulting reduced order optimized airframe subcontroller is

$$SK_{Ared} = \begin{bmatrix} 0 & 1.00E^{+0} & 0 & 0 & 0 & 0 & 0 & -2.81E^{-1} & 8.98E^{-2} \\ -2.84E^{+0} & -4.39E^{+1} & 0 & 0 & 0 & 0 & 0 & -1.39E^{-2} & 1.18E^{-1} \\ 0 & 0 & 0 & 1.00E^{+0} & 0 & 0 & 1.29E^{+0} & -5.50E^{+0} \\ 0 & 0 & -1.71E^{+3} & -6.19E^{+1} & 0 & 0 & -1.53E^{+1} & -9.18E^{+2} \\ 0 & 0 & 0 & 0 & 0 & 1.00E^{+0} & 1.38E^{+0} & -1.20E^{-1} \\ 0 & 0 & 0 & 0 & -6.92E^{+1} & -1.18E^{+1} & -5.03E^{+0} & 5.95E^{-1} \\ -1.87E^{-1} & -7.05E^{+0} & 7.83E^{+0} & 7.95E^{-1} & 3.43E^{+1} & 1.28E^{+0} & -3.24E^{+0} & 5.62E^{-1} \\ -1.95E^{+0} & 1.40E^{+1} & 2.99E^{-2} & 8.66E^{-4} & -8.63E^{+0} & -2.28E^{+0} & 1.60E^{-3} & -2.25E^{-3} \end{bmatrix}$$

•			
_			
•			
	-		

### REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND	DATES COVERED				
,	March 1994	Fi	nal Contractor Report				
4. TITLE AND SUBTITLE			5. FUNDING NUMBERS				
Computer Code for Control							
A User's Manual							
A User's Manual							
			WILL FOE 62 50				
6. AUTHOR(S)	WU-505-62-50						
	C-NAG3-11467						
Phillip H. Schmidt and Asin							
7. PERFORMING ORGANIZATION N	AME(S) AND ADDRESS(ES)		8. PERFORMING ORGANIZATION				
		Ì	REPORT NUMBER				
University of Akron							
Akron, Ohio 44325			E-8654				
ARION, OIDO 44323			2 0051				
9. SPONSORING/MONITORING AGE	NCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER				
37.7. 14 10	A di-i		ACERO, RELON, NUMBER				
National Aeronautics and S	pace Administration		X1.01.00 107001				
Lewis Research Center	,		NASA CR-195291				
Cleveland, Ohio 44135-3	191						
11. SUPPLEMENTARY NOTES							
Project Manager, Sanjay Ga	arg, Instrumentation and Control	Technology Division, o	rganization code 2550, NASA				
Lewis Research Center, (21	6) 433–2355.						
12a. DISTRIBUTION/AVAILABILITY	STATEMENT		12b. DISTRIBUTION CODE				
	<del> </del>						
Unclassified - Unlimited							
Subject Categories 08 and 6	63						
			-				
13. ABSTRACT (Maximum 200 word	(s)						
15. ABSTITAGE (MEXIMUM 250 WOLD	,						
This is a user's manual for	the computer code for partitioning	g a centralized controlle	er into decentralized subcontrollers				
with applicability to Integral	ated Flight/Propulsion Control (II	EDC) Partitioning of a c	centralized controller into two				
with applicability to integra	aled Fight/Flopulsion Condot (if	ada is based is discussed	d. The algorithm uses parameter				
subcontrollers is described	and the algorithm on which the c	oue is based is discussed	d. The algorithm uses parameter				
optimization of a cost funct	tion which is described here. The	major data structures at	nd functions are described. Specific				
instructions are given. The	user is led through an example of	an IFPC application.					
			15. NUMBER OF PAGES				
14. SUBJECT TERMS			15. NUMBER OF PAGES				
Integrated control; Centrali	zed control; Decentralized control	ol; Flight control;	16. PRICE CODE				
Propulsion control			A04				
		44 0501000/014000004					
17. SECURITY CLASSIFICATION	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICA OF ABSTRACT	ION 20. LIMITATION OF ADSTRACT				
OF REPORT	Unclassified	Unclassified					
Unclassified	Officiassified	Unclassified					